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

(10) **DE 602 21 861 T2** 2008.03.06

(12) # Übersetzung der europäischen Patentschrift

(97) **EP 1 430 680 B1**
(21) Deutsches Aktenzeichen: **602 21 861.6**
(86) PCT-Aktenzeichen: **PCT/GB02/04004**
(96) Europäisches Aktenzeichen: **02 755 301.5**
(87) PCT-Veröffentlichungs-Nr.: **WO 2003/021908**
(86) PCT-Anmeldetag: **02.09.2002**
(87) Veröffentlichungstag der PCT-Anmeldung: **13.03.2003**
(97) Erstveröffentlichung durch das EPA: **23.06.2004**
(97) Veröffentlichungstag der Patenterteilung beim EPA: **15.08.2007**
(47) Veröffentlichungstag im Patentblatt: **06.03.2008**

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

(30) Unionspriorität:
0121299 **03.09.2001** **GB**

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

(73) Patentinhaber:
Intercede Ltd., Lutterworth, Leicestershire, GB

(74) Vertreter:
Strehl, Schübel-Hopf & Partner, 80538 München

(72) Erfinder:
EDWARDS, Christopher 16 Kingsdown Mount, Nottingham NG8 2RQ, GB; WARD, Christopher Robert 26 Be, Nottingham NG12 4ED, GB

(54) Bezeichnung: **SERVER MIT DATEIVERIFIKATION**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

[0001] Diese Erfindung betrifft das Antworten auf Informationsanfragen und insbesondere, jedoch nicht ausschließlich, die Verwendung von Servern zum automatischen Antworten auf Informationsanfragen auf der Grundlage von Informationen, die in Dateien gespeichert sind, die einem Prozessor des Servers zugänglich sind.

[0002] Solche Server werden gewöhnlich in privaten und öffentlichen Netzen in einer Vielzahl von Zusammenhängen verwendet und sind beispielsweise aus EP 106975 A1 bekannt.

[0003] Im Fall eines Internet-Web-Servers können die in einer Antwort enthaltenen Informationen als eine Web-Seite präsentiert werden, die statisch sein kann oder dynamisch erzeugt werden kann, wobei sie typischerweise durch eine Anwendung erzeugt wird, die mit einer oder mehreren gespeicherten Dateien arbeitet. Die Antwort kann auch Web-Seiten einschließen, welche Formularausfüllungsfähigkeiten bereitstellen, die während kommerzieller Transaktionen oder während des Austausches vertraulicher Informationen verwendet werden.

[0004] Es ist bekannt, dass solche Server Angriffen offen stehen, die darauf abzielen, von dem Server bereitgestellte Web-Seiten in bösartiger Weise zu verändern. Angreifer können auch die betrügerische Absicht haben, in dem Server vorhandene vertrauliche Informationen freizugeben oder betrügerische Transaktionen verschiedener Arten, einschließlich jener, die auf dem Ändern eines in einer Web-Seite definierten Formulars in einer solchen Weise, dass ein Kunde so antwortet, dass er dem Angreifer vertrauliche Informationen offen legt, beruhen, zu bewirken.

[0005] Solche Angriffe zielen gemeinhin darauf ab, die im Web-Server gespeicherten Dateien so zu modifizieren, dass die erzeugten Web-Seiten so modifiziert werden, dass sie den Absichten des Angreifers genügen. Firewall-Systeme werden üblicherweise verwendet, um solche Server zu schützen, und sie können, abhängig von ihrer Komplexität, verhältnismäßig erfolgreich sein, um die meisten Angreifer abzuschrecken. Solche Firewall-Systeme sind jedoch nicht undurchdringlich und können nicht vor Angriffen von jenen schützen, die einen privilegierten Zugang zu der durch die Firewall geschützen Umgebung haben.

[0006] Es ist möglich, die Dateien in einer Form zu speichern, in der ein Prozessor des Servers auf einer Nurlesebasis Zugang hat. Eine Anzahl von Vorrichtungen bietet diese Möglichkeit, wie CD-ROM-Laufwerke, bei denen das Speichermedium nicht überschrieben werden kann, oder Vorrichtungen, bei denen ein Speichermedium, das überschrieben werden kann, durch eine elektronische Logik in einem Speichersystem geschützt ist, das dadurch in die Lage versetzt wird, auf einer Nurlesebasis Zugang zu erhalten.

[0007] Das Problem, das bei solchen Nurlesespeichern auftritt, die zur Massenspeicherung von Dateien geeignet sind, besteht darin, dass der Zugriff auf die Dateien typischerweise langsamer ist als die Zugriffszeit, die verfügbar ist, wenn ein Prozessor Zugang zu Dateien in einem Direktzugriffsspeicher in der Art eines Speicherschaltungsfelds oder eines Festplattenlaufwerks hat. Es ist im Allgemeinen bei solchen Speichern mit einem schnellen Zugriff nicht praktikabel, zuverlässig zu verhindern, dass Dateien überschrieben werden.

[0008] Es ist auch bekannt, eine Validierung der Integrität gespeicherter Dateien durch Berechnen einer Validierungsfunktion der Dateiinhalte auszuführen, um einen Sicherheitswert zu erhalten, der dann mit einem Referenzwert verglichen werden kann, welcher zuvor durch ein ähnliches Verfahren, beispielsweise als die Datei erzeugt wurde, berechnet wurde. Jeder Unterschied zwischen dem Sicherheitswert und dem Referenzwert liefert daher einen Hinweis, dass die Dateiinhalte geändert worden sind.

[0009] Ein Problem, das bei solchen Validierungsverfahren auftritt, besteht darin, dass ein Angreifer nicht nur die Datei ändern kann, sondern auch auf die Validierungsfunktion zugreifen kann, um den neuen Sicherheitswert für die beschädigte Datei zu berechnen, und den Referenzwert dementsprechend ändern kann. Bei einer anschließenden Berechnung des Sicherheitswerts wird dann die Beschädigung der Datei nicht festgestellt, weil der Sicherheitswert und der (geänderte) Referenzwert übereinstimmen.

[0010] Das vorstehende Problem wird durch ein Verfahren, eine Vorrichtung und ein Speichermedium nach den Ansprüchen 1, 17 und 33 gelöst.

[0011] Die vorliegende Erfindung strebt an, eine verbesserte Vorrichtung und ein verbessertes Verfahren zum Antworten auf Informationsanfragen bereitzustellen.

[0012] Eine Ausführungsform der vorliegenden Erfindung sieht ein Verfahren und eine Vorrichtung vor, wobei Dateien in einem Server in schnell zugänglicher Form gespeichert werden und die Dateiinhalte verifiziert werden, wenn eine Antwort auf eine Informationsanfrage erzeugt wird.

[0013] Bevorzugte Ausführungsformen der vorliegenden Erfindung werden nun nur als Beispiel mit Bezug auf die anliegende Zeichnung beschrieben, wobei:

[0014] Fig. 1 eine Schemazeichnung eines Servers in Zusammenhang mit dem Bereitstellen von Web-Seiten über das Internet ist,

[0015] Fig. 2 eine Schemazeichnung ist, die weitere Einzelheiten des Servers aus Fig. 1 zeigt,

[0016] Fig. 3 ein Flussdiagramm ist, das Schritte des Prozesses zum Erzeugen einer Web-Seite zeigt,

[0017] Fig. 4 ein Flussdiagramm ist, das den Dateiverifikationsschritt aus Fig. 3 detailliert zeigt,

[0018] Fig. 5 ein Flussdiagramm ist, das einen Dateisignaturprozess zeigt,

[0019] Fig. 6 ein schematisches Diagramm ist, das die Datenstruktur im RAM gespeicherter Daten zeigt,

[0020] Fig. 7 ein Diagramm ist, das dem Server bereitgestellte Software zeigt,

[0021] Fig. 8 ein schematisches Diagramm ist, das den Datenfluss in dem Verifikationsprozess zeigt, und

[0022] Fig. 9 ein schematisches Diagramm ist, das Verfahren zum Bereitstellen von Programmen für den Server zeigt.

[0023] Fig. 1 zeigt schematisch einen Web-Server **1** zum Antworten auf Anfragen von Client-Endgeräten **2** über das Internet **3**. Die Hardwarestruktur des Servers **1**, einschließlich eines Prozessors **20**, ist in Fig. 2 dargestellt und wird nachstehend beschrieben. Eine Firewall **4** stellt eine Sicherheitsmaßnahme gegen einen äußeren Angriff durch bösartige Angreifer über das Internet **3** auf den Web-Server **1** bereit, indem sie als ein Proxy-Server wirkt, der das Durchsenden von Kommunikationen nur nach dem Überprüfen ihrer Authentizität zulässt. Wenngleich die Firewall **4** in Fig. 1 als ein einziges Element dargestellt ist, umfasst sie typischerweise ein System aus mehreren Komponenten, welche durch ein öffentliches oder privates Netz miteinander verbunden werden können.

[0024] Der Web-Server **1** empfängt Anfragenachrichten vom Client-Endgerät **2** über eine Anfrageeingabeschnittstelle **5**, welche die Anfrage über ein Filter **7** an eine Web-Seiten-Erzeugungseinrichtung **6** weiterleitet.

[0025] Durch die Web-Seiten-Erzeugungseinrichtung **6** erzeugte Web-Seiten werden als Antwortnachrichten durch eine Antwortausgabeschnittstelle **8** ausgegeben, um sie über das Internet **3** zum Client-Endgerät **2** weiterzuleiten.

[0026] Eine Sammlung von Dateien wird in einem ersten Speicher **9** gespeichert, der in der Hinsicht die Eigenschaften eines Direktzugriffsspeichers (RAM) aufweist, dass er das schnelle Lesen von Informationen aus dem Speicher und das schnelle Schreiben von Informationen in den Speicher erleichtert, wodurch das schnelle Abrufen von Dateien durch die Web-Seiten-Erzeugungseinrichtung **6** erleichtert wird.

[0027] Kopien der im ersten Speicher **9** gespeicherten Dateien werden auch als Vorlagedateien in einem zweiten Speicher **10** gespeichert, der die Eigenschaften eines Nurlesespeichers (ROM) aufweist, so dass die Web-Seiten-Erzeugungseinrichtung **6** und ihre zugeordneten Komponenten nur lesend auf die Vorlagedateien zugreifen können. In dem Beispiel aus Fig. 1 liegt der zweite Speicher **10** in Form einer CD-ROM vor, wobei die CD ein nicht wiederbeschreibbares Speichermedium darstellt.

[0028] Das Filter **7** fängt eingehende Anfragenachrichten, welche über die Anfrageeingabeschnittstelle **5** empfangen wurden, ab und gibt die Anfragenachrichten erst nach Beendigung eines Validierungsvorgangs an die Web-Seiten-Erzeugungseinrichtung **6** ab. Die Anfrageeingabeschnittstelle **5** identifiziert anhand des Kontexts der Anfrage eine Liste von der Web-Seiten-Erzeugungseinrichtung **6** benötigter Dateien für das Erzeugen der Antwortnachricht, d.h. der Web-Seite. Diese Liste von Dateien wird durch das Filter **7** zu einer Integritätsprüfkomponente **11** weitergeleitet, welche für jede aufgelistete Datei die Integrität der im ersten Speicher **9** gespeicherten Datei prüft. Wenn die Integrität aller Dateien in der Liste bestätigt wurde, wird dies dem Filter **7** mitgeteilt, der dann die Anfragenachricht für die Verarbeitung durch die Web-Seiten-Erzeugungseinrichtung **6** unter Verwendung der aufgelisteten Dateien abgibt.

[0029] Falls die Integritätsprüfkomponente **11** feststellt, dass die Inhalte irgendwelcher der aufgelisteten Dateien verfälscht sind, wird die Identität der Datei einer Reparaturkomponente **12** mitgeteilt, welche eine entsprechende Vorlagedatei aus dem zweiten Speicher **10** abruft und die Vorlagedatei verwendet, um die Inhalte des ersten Speichers **9** zu reparieren, indem die verfälschten Inhalte gelöscht werden und sie durch Inhalte ersetzt werden, die aus der im zweiten Speicher gespeicherten Vorlagedatei kopiert werden.

[0030] Die Integritätsprüfkomponente **11** ist dann in der Lage, die Integrität der Datei zu bestätigen, und der Vorgang der Web-Seiten-Erzeugung kann dann fortgesetzt werden.

[0031] Ein Vorteil des vorstehend erwähnten Web-Servers **1** besteht darin, dass der erste Speicher **9** so ausgewählt werden kann, dass er eine verhältnismäßig schnelle Antwortzeit aufweist, wodurch

die schnelle Verarbeitung der eingegebenen Anfragenachrichten gewährleistet ist. In dem Fall, in dem eine Dateiverfälschung entdeckt wird, wird der Vorgang jedoch notwendigerweise dadurch verlangsamt, dass es notwendig ist, auf die jeweilige Vorlagedatei Bezug zu nehmen, die sich im zweiten Speicher **10** befindet, der eine verhältnismäßig langsame Antwortzeit hat, und dass es notwendig ist, die Reparatur auszuführen, bevor fortgefahren wird. Ein anschließender schneller Dienst beim Antworten auf weitere Anfragen ist dann wieder aufnehmbar.

[0032] Der Web-Server **1** wird dadurch im Wesentlichen immun gegen Angriffe gemacht, weil alle Informationen, die durch den Server **1** bereitgestellt werden, am Auslieferungspunkt verifiziert werden.

[0033] Ein Alarmerzeugungsmodul **13** ist auch bereitgestellt und eingerichtet, um ein Alarmsignal von der Integritätsprüfkomponente **11** in dem Fall zu empfangen, dass eine verfälschte Datei entdeckt wurde. Der Alarm kann in Form einer Benachrichtigungsausgabe an einen Bediener vorliegen oder in Form eines Signals für ein Steuersystem vorliegen, das einen Bericht erzeugen kann und automatisch weitere berichtigende Tätigkeiten, wie eine systematische Prüfung der Validität aller im ersten Speicher **9** enthaltener Dateien, einleiten.

[0034] Eine Datenwartungskomponente **14** ist in der Lage, Daten in den zweiten Speicher **10** zu schreiben, beispielsweise wenn eine neue Datei erzeugt wird. In dem vorliegenden Beispiel weist die Datenwartungskomponente **14** Software und Hardware zum Schreiben von Daten in eine nicht wiederbeschreibbare CD auf. Falls eine existierende Datei eine Aktualisierung benötigt, um verbesserte Daten aufzunehmen, während ihre ursprüngliche Dateikennung beibehalten wird, ist es erforderlich, das Aufzeichnungsmedium, d.h. die CD, zu ersetzen, indem eine neue CD bereitgestellt wird, auf der alle Dateien neu aufgezeichnet werden.

[0035] Die Integritätsprüfkomponente **11** ist in der Lage, die Integrität einer im ersten Speicher **9** enthaltenen Datei zu prüfen, indem sie eine Verifikationsfunktion berechnet, welche auf die Dateiinhalte angewendet wird, um einen berechneten Sicherheitswert zu erhalten. Im vorliegenden Beispiel ist die Funktion eine Standard-SHA1-Hash-Funktion, deren Ergebnis 160 Bytes an Daten sind, welche den Sicherheitswert bilden. Für jede im ersten Speicher **9** gespeicherte Datei wird auch eine jeweilige Signatur gespeichert, welche durch eine Signierkomponente **15** zur Zeit der Dateieingabe in den Server **1** erzeugt wird, wobei die Signatur dadurch erzeugt wird, dass zuerst die Verifikationsfunktion auf die Dateiinhalte angewendet wird und dann der sich ergebende Sicherheitswert unter Verwendung eines privaten Schlüssels verschlüsselt wird, um die Signatur zu erhalten. Während dieses Verschlüsselungsprozesses muss eine Person, die einen privilegierten Zugang hat, wie ein Systemadministrator, einer Vorrichtung **17** zum Zugang zum privaten Schlüssel des Web-Servers **1** einen tragbaren privaten Schlüsselträger **16** präsentieren, um zu ermöglichen, dass der private Schlüssel **19** gelesen und in die Signierkomponente **15** eingegeben wird. Im vorliegenden Beispiel ist der tragbare private Schlüsselträger **16** eine Chipkarte und ist die Vorrichtung **17** zum Zugriff auf den privaten Schlüssel ein Chipkartenleser. Der tragbare private Schlüsselträger **16** wird unmittelbar nach dem Signierprozess entfernt, so dass er dem Prozessor **20** und anderen Benutzern des Web-Servers **1** nicht mehr zur Verfügung steht. Jeder Angreifer, der Zugang zum Prozessor **20** erhält, kann daher den privaten Schlüssel **19** nicht erhalten, und es wird daher verhindert, dass er Signaturen erzeugt, welche den Verifikationsprozess besiegen können.

[0036] Der Verifikationsprozess wird nachstehend in weiteren Einzelheiten beschrieben.

[0037] Fig. 2 zeigt schematisch die den Web-Server **1** ausmachende Hardware, einschließlich des Prozessors **20**, der über einen Datenbus **21** mit dem ersten Speicher **9** und dem zweiten Speicher **10** verbunden ist. Mit dem Datenbus **21** sind auch ein Programmspeicher **22** (typischerweise ein Festplattenlaufwerk), um die für das Betreiben des Servers erforderlichen Programme zu speichern, ein Arbeitsspeicher **23** (RAM), eine Netzwerkschnittstelle **24**, eine Chipkartenleser-Schnittstelle **25** und eine graphische Benutzerschnittstelle (GUI) **26** verbunden.

[0038] Fig. 3 zeigt die Schritte des Verfahrens, das vom Web-Server **1** bei der Verarbeitung einer vom Client-Endgerät **2** empfangenen Anfrage ausgeführt wird.

[0039] In Schritt **30** empfängt der Web-Server **1** eine Anfrage über die Firewall **4**, wobei die Anfragenachricht durch Browser-Software des Client-Endgeräts **2** eingeleitet wurde und unter Verwendung der geeigneten URL der Web-Seite an den Web-Server adressiert wurde. In Schritt **30** analysiert die Anfrageeingabeschnittstelle **5** die Anfragenachricht und identifiziert die Liste der Dateien, die zum Erzeugen einer geeigneten Antwort erforderlich sind.

[0040] In Schritt **31** wird die Anfrage vom Filter **7** aufgefangen, und in Schritt **32** ruft die Integritätsprüfkomponente **11** die erste der benötigten Dateien aus dem ersten Speicher **9** ab.

[0041] In Schritt **33** führt die Integritätsprüfkomponente **11** den Dateiverifikationsprozess an der Datei aus, und in Schritt **34** bestimmt sie, ob die Integrität der Datei verifiziert wurde. Falls das Ergebnis darin besteht, dass die Integrität bestätigt wird, wird in

Schritt **35** festgestellt, ob noch weitere aufgelistete Dateien verifiziert werden müssen, und der Prozess beginnt von Schritt **32** an neu, falls dies der Fall ist.

[0042] Falls in Schritt **34** das Ergebnis negativ ist, d.h. festgestellt wird, dass die Datei verfälscht ist, wird ein Alarm-Hinweiszeichen in Schritt **36** gesetzt, so dass nach Beendigung des Verifikationsprozesses die erforderlichen Alarmnachrichten und Steuertätigkeiten erzeugt werden.

[0043] In Schritt **37** wird eine Vorlagedatei, die die gleiche Kennung aufweist wie die verfälschte Datei, zusammen mit der gespeicherten Signatur der Vorlagedatei aus dem zweiten Speicher **10** ausgelesen, und in Schritt **38** wird die verfälschte Datei repariert, indem sie im ersten Speicher **9** durch eine Kopie der abgerufenen Vorlagedatei ersetzt wird. Die im ersten Speicher **9** gespeicherte Signatur, die der verfälschten Datei entspricht, wird auch repariert, indem sie durch die Signatur überschrieben wird, die aus dem zweiten Speicher **10** abgerufen wurde.

[0044] Schritt **35** folgt dann, in dem festgestellt wird, ob weitere Dateien verbleiben.

[0045] Sobald die Integrität aller Dateien verifiziert wurde und alle notwendigen Reparaturen beendet wurden, folgt Schritt **39**, in dem die Antwortnachricht unter Verwendung der verifizierten Dateien erzeugt wird.

[0046] Der Dateiverifikationsschritt **33** ist in Fig. 4 dargestellt.

[0047] In Fig. 4 werden in Schritt **40** die Inhalte der Datei aus dem ersten Speicher **9** ausgelesen, und in Schritt **41** wird auch das Datenfeld, das die Signatur der Datei enthält, aus dem ersten Speicher **9** ausgelesen.

[0048] In Schritt **42** wird die in Schritt **41** gelesene Signatur unter Verwendung eines öffentlichen Schlüssels entschlüsselt, der dem Prozessor **20** zur Verfügung steht, um einen Referenzwert zu erhalten. Dieser Referenzwert gleicht dem Ergebnis, das erhalten wurde, als die Verifikationsfunktion bei der Eingabe der Datei in den Server **1**, wenn die Signierkomponente **15** den Referenzwert verschlüsselt hat, um die Signatur zu erhalten, an den Dateiinhalten ausgeführt wurde.

[0049] In Schritt **43** wird die Verifikationsfunktion dann auf die aus dem ersten Speicher **9** ausgelesenen Inhalte der Datei angewendet, um einen Sicherheitswert zu erhalten.

[0050] In Schritt **44** wird der Sicherheitswert mit dem Referenzwert verglichen, und falls sie gleich sind, wird die Integrität der Datei bestätigt, und in Schritt **45** wird ein Hinweiszeichen gesetzt, um die folgenden Verarbeitungsschritte darauf hinzuweisen, dass die Integrität verifiziert wurde.

[0051] Falls der Sicherheitswert jedoch nicht gleich dem Referenzwert ist, wird dies als ein Hinweis darauf angenommen, dass die Datei beschädigt ist, und in Schritt **46** wird ein Hinweiszeichen gesetzt, um anzugeben, dass die Datei beschädigt ist, wodurch die Reparaturprozedur aufgerufen wird.

[0052] Fig. 5 zeigt den Prozess des Signierens der Datei, welcher von der Signierkomponente **15** immer dann ausgeführt wird, wenn eine Datei neu in den Web-Server **1** eingegeben wird.

[0053] In Schritt **50** werden die Inhalte der Datei verfasst oder auf andere Weise in den Web-Server **1** importiert. In Schritt **51** wird die Verifikationsfunktion auf die Inhalte der Datei angewendet, um einen Zahlenwert zu erhalten, wobei der Zahlenwert im vorliegenden Beispiel ein Hash-Wert ist.

[0054] Ein Systemadministrator präsentiert in Schritt **52** den tragbaren Schlüsselträger **16** der privaten Schlüsselzugangsvorrichtung **17**, was im vorliegenden Beispiel darin besteht, dass einem Chipkartenleser eine Chipkarte präsentiert wird. Der private Schlüssel für den Verschlüsselungsprozess wird aus dem tragbaren privaten Schlüsselträger **16** (Chipkarte) gelesen und in Schritt **53** verwendet, um, den in Schritt **51** erhaltenen Referenzwert zu verschlüsseln.

[0055] In Schritt **54** wird der tragbare Schlüsselträger **16** aus der Vorrichtung **17** zum Zugriff auf den privaten Schlüssel entfernt.

[0056] In Schritt **55** wird der verschlüsselte Referenzwert zusammen mit den Inhalten der Datei als eine Dateisignatur im ersten Speicher **9** gespeichert. In Schritt **56** werden die Signatur und die Dateiinhalte im zweiten Speicher **10** gespeichert, um eine Vorlagekopie der Datei zu erzeugen, die in einem anschließenden Reparaturprozess verwendbar ist.

[0057] Fig. 6 zeigt ein Beispiel im ersten Speicher **9** und im zweiten Speicher **10** gespeicherter Dateien. Eine erste Datei **60** weist Inhalte **61** im HTML-(HyperText Markup Language)-Format, die bei der Erzeugung der Web-Seite verwendbar sind, und einen Dateikopfabschnitt **62**, der ein Datenfeld aufweist, in dem die Signatur gespeichert ist, auf.

[0058] Der Begriff "Inhalt" bzw. "Inhalte" wird im vorliegenden Zusammenhang verwendet, um den Hauptteil des die Datei bildenden Codes mit Ausnahme eines Kopfabschnitts der Datei, der die Signatur enthält, anzugeben, so dass in der vorhergehenden Beschreibung Bezugnahmen auf die Anwendung der

Verifikationsfunktion auf die Inhalte einer Datei so verstanden werden sollten, dass die Funktion auf jene Inhalte **61** der Datei angewendet wird, welche den Kopf **62** nicht aufweisen.

[0059] Eine zweite Datei **63** enthält Inhalte **64** im ASP-(Active Server pages)-Format, die in Zusammenhang mit der HTML-Datei **60** bei der Präparation der Web-Seite verwendbar sind. Die zweite Datei **63** weist ähnlich einen Kopf **65** auf, der die Signatur enthält, die während des Signierprozesses durch Verschlüsseln des Referenzwerts auf der Grundlage der Inhalte **64** erhalten wird.

[0060] Eine dritte Datei **66** enthält Inhalte **67** in Form von XML-Style-Sheets (XSL), die in Zusammenhang mit der ersten Datei **60** und der zweiten Datei **63** bei der Erzeugung der Web-Seite verwendbar sind. Die dritte Datei **66** weist auch einen Kopfabschnitt **68** auf, der eine jeweilige Signatur enthält.

[0061] Eine vierte Datei **69** enthält Daten, welche ein Bild im GIF-(Graphics Interchange Format)-Format darstellen. Weil dieser Dateityp keinen Kopfabschnitt vorsieht, muss die jeweilige Signatur der vierten Datei **69** getrennt gespeichert werden und ist in einer Nachschlagetabelle **690** enthalten, die im ersten Speicher **9** gespeichert ist und eine Registrierung zwischen Dateikennungen **691** und ihren jeweiligen Signaturen **692** bereitstellt. Jede Dateikennung **691** definiert eine Speicheradresse der Datei, auf die sich die entsprechende Signatur **692** bezieht.

[0062] Eine entsprechende Datenstruktur, welche Dateien und eine Nachschlagetabelle enthält, ist im zweiten Speicher **10** gespeichert.

[0063] Fig. 7 zeigt schematisch die im Programmspeicher **22** aus Fig. 2 enthaltenen Softwarekomponenten.

[0064] Ein IIS (Internet Information Server) **70** weist eine ISAPI (Internet Server Application Programming Interface) **71** auf, welche konfiguriert ist, um die Filterkomponente **7**, die Integritätsprüfkomponente **11** und die Reparaturkomponente **12** zu definieren. Die Integritätsprüfkomponente **11** weist Entschlüsselungssoftware **72** zum Entschlüsseln von Signaturen auf.

[0065] Die Datenwartungskomponente **14** ist auch zusammen mit der Signierkomponente **15** bereitgestellt, welche Verschlüsselungssoftware **73** aufweist.

[0066] Es ist auch ein Alarmerzeugungsmodul **13** enthalten. Fig. 8 zeigt eine zweckmäßige Zusammenfassung des Datenflusses in dem vorstehend mit Bezug auf Fig. 4 beschriebenen Verifikationsprozess in dem Fall, in dem die erste Datei **60** aus Fig. 6 der Gegenstand des Verifikationsprozesses ist.

[0067] Wie in Fig. 8 dargestellt ist, werden die Dateiinhalte **61** in die Verifikationsfunktion **80** eingegeben, deren Rechenergebnis ein Sicherheitswert **81** ist.

[0068] Die aus dem Dateikopf der Datei **60** gelesene Signatur **62** wird unter Verwendung von Entschlüsselungssoftware **72** und des im ersten Speicher **9** gespeicherten öffentlichen Schlüssels **82** entschlüsselt, um einen Referenzwert **83** zu erhalten.

[0069] Ein Vergleicher **84** vergleicht dann den Sicherheitswert **81** und den Referenzwert **83** und setzt das Datei-gültig-Hinweiszeichen **85**, falls die Werte identisch sind. Falls die Werte nicht identisch sind, setzt der Vergleicher **84** das Datei-verfälscht-Hinweiszeichen **86** und das Alarm-Hinweiszeichen **87**.

[0070] In der vorstehenden Beschreibung wird auf die Verschlüsselung unter Verwendung eines privaten Schlüssels **19** und die Entschlüsselung unter Verwendung eines öffentlichen Schlüssels **82** Bezug genommen. Solche Verschlüsselungssoftware, welche Verschlüsselungs- und Entschlüsselungsalgorithmen aufweist, ist im Handel erhältlich und beruht typischerweise auf den Eigenschaften einer Modularithmetik, welche in einem Galois-Feld angewendet wird. Für den Zweck der vorstehend offenbarten Ausführungsformen ist es für eine unter Verwendung eines ersten numerischen Schlüssels verschlüsselte Zahl ausreichend, dass sie unter Verwendung eines zweiten numerischen Schlüssels, der sich in einer Art auf den ersten numerischen Schlüssel bezieht, die nicht durch einen Angreifer festgestellt werden kann, entschlüsselbar ist. Der erste numerische Schlüssel wurde vorstehend als privater Schlüssel bezeichnet, wodurch impliziert wird, dass die Kenntnis des Schlüssels nur jenen zur Verfügung steht, die privilegierten Zugang zum System haben, während der als der öffentliche Schlüssel **82** bezeichnete zweite numerische Schlüssel im ersten Speicher **9** vorhanden sein kann und dem Prozessor **20** zugänglich sein kann. Ein Angreifer, der Zugang zum Prozessor erhält, kann Zugang zum öffentlichen Schlüssel **82** gewinnen und, falls er auch Zugang zur Verifikationsfunktion **80** erhält, die Signatur **62** einer im ersten Speicher **9** gespeicherten Datei **60** entschlüsseln. Falls der Angreifer jedoch die Dateiinhalte **61** modifiziert, hat er keine Möglichkeit, einen neuen Wert der Signatur **62** zu bestimmen, die, wenn sie durch die Integritätsprüfkomponente **11** entschlüsselt wurde, gleich dem Sicherheitswert **81** ist, der durch die Verifikationsfunktion **80** auf der Grundlage der modifizierten Dateiinhalte berechnet wurde. Dies liegt daran, dass der Angreifer keinen Zugang zum privaten Schlüssel **19** haben kann, der erforderlich ist, um die richtige Verschlüsselung zum Erhalten der Signatur auszuführen.

[0071] Es sind verschiedene Anwendungen des

vorstehend beschriebenen Web-Servers **1** vorgesehen, einschließlich beispielsweise eines Chipkarten ausgebenden Systems, bei dem in dem ersten Speicher **9** enthaltene Datendateien für das Ausgeben von Chipkarten verwendet werden und vertrauliche Codes und Informationen enthalten. Der Prozess des Ausgebens einer Chipkarte über das Internet erfordert es, dass eine Formularausfüllfunktion ausgeführt wird, so dass persönliche Daten von Kunden dem System bereitgestellt werden können. Der Server **1** muss daher in diesem Beispiel Antwortnachrichten für das Präsentieren von Web-Seiten, die diese Formulare enthalten, erzeugen. Die Sicherheit, die durch die vorstehenden Ausführungsformen bereitgestellt wird, gewährleistet, dass ein Angreifer nicht die Web-Seiten modifizieren kann, welche solche Formulare enthalten, die dem Kunden präsentiert werden, und sie macht daher ein betrügerisches Abzweigen von Informationen oder ein Austauschen von Informationen in der Web-Seite, um den Kunden dazu zu bringen, Informationen herauszugeben, die dem Angreifer zugeführt werden, unmöglich.

[0072] Fig. 9 zeigt schematisch die Art, in der das für den Betrieb des Web-Servers benötigte Programm installiert werden kann. In einem Speichermedium **90** gespeicherte Programme können in ein Lesegerät der Web-Server-Vorrichtung **1** eingegeben werden. Alternativ können Programme über ein Netzwerk in der Art des Internets **3** in Form elektronischer Signale **91** empfangen werden.

[0073] Ein Aspekt der vorliegenden Erfindung sieht demgemäß ein Speichermedium **90** vor, welches von einem Prozessor implementierbare Befehle für das Steuern eines Prozessors zur Ausführung des vorstehend beschriebenen Verfahrens speichert. Zusätzlich ist gemäß einem anderen Aspekt der vorliegenden Erfindung ein elektrisches Signal **91** vorgesehen, welches vom Prozessor implementierbare Befehle überträgt, um einen Prozessor zu steuern, damit er das vorstehend beschriebene Verfahren ausführt.

[0074] Es sind auch alternative Ausführungsformen innerhalb des Schutzumfangs der vorliegenden Erfindung vorgesehen. Während das Beispiel aus Fig. 1 beispielsweise unter Verwendung eines einzigen Prozessors funktioniert, kann der Web-Server **1** auch unter Verwendung mehrerer Prozessoren arbeiten. Insbesondere kann die in Fig. 1 durch die gepunktete Linie **18** dargestellte Unterteilung eine Unterteilung der von zwei getrennten Prozessoren ausgeführten Steuerung bezeichnen, wobei die Signier- und Datenwartungsaufgaben durch einen zweiten Prozessor (nicht dargestellt) ausgeführt werden. Die Komponenten **17**, **15**, **14** und **10** können sich fern vom Rest des Web-Servers **1** befinden und damit über ein privates oder öffentliches Netzwerk unter Verwendung geeigneter Firewalls und Sicherheitsmaßnahmen verbunden werden.

[0075] Die bevorzugte Ausführungsform wurde mit Bezug auf die Verwendung einer Chipkarte als privater Schlüsselträger **16** beschrieben. Es sind andere Formen des Trägers **16**, einschließlich eines HSM (Hardware Security Module), das eine Festplattenvorrichtung mit einem internen Prozessor zum Bereitstellen eines sicheren verschlüsselten Datenspeichers aufweist, vorgesehen. Das HSM ist transportierbar und kann, durch eine Systemsteuereinrichtung gesteuert, sicher in einer Entfernung vom Web-Server **1** gespeichert werden.

[0076] Andere Formen des tragbaren privaten Schlüsselträgers **16** können einfach Datenträger sein, die keine Verarbeitungsfähigkeit haben, und optisch codierte Karten oder andere geeignete Speichermedien einschließen.

[0077] Optional kann die Integrität in dem ersten Speicher **9** enthaltener Dateien durch ein Dateiwartungsprogramm systematisch geprüft werden. Diese Prozedur kann beispielsweise zeitlich so festgelegt werden, dass sie mit Perioden zusammenfällt, in denen der Web-Server **1** in Ruhe ist oder nicht bei der vollen Kapazität verwendet wird.

[0078] In der vorstehenden Beschreibung sind die Inhalte der gespeicherten Dateien typischerweise Daten, die in Anwendungen zur Erzeugung einer Web-Seite verwendet werden. Die Dateien können ähnlich Computerprogramme zur Verwendung bei der Web-Seiten-Erzeugung, beispielsweise kompilierte Programme in ausführbarer Form oder Programme zur Eingabe in eine Arbeitsumgebung während der Web-Seiten-Erzeugung, enthalten, und der Begriff "Dateiinhalte" sollte in weitem Sinne so ausgelegt werden, dass er solche Programme einschließt, sofern es geeignet ist.

[0079] In der vorstehenden Beschreibung sind die Namen IIS und ISAPI Warenzeichen.

Patentansprüche

1. Verfahren zum Beantworten einer Informationsanfrage, wobei in dem Verfahren
wenigstens ein Prozessor (**20**) betrieben wird, um die folgenden Schritte auszuführen:
Erzeugen (**39**) geforderter Information unter Verwendung von Inhalten wenigstens einer Datei (**60**, **63**, **66**, **69**), die in einem ersten Speicher (**9**) gespeichert ist, der für den Prozessor zum Lesen und Schreiben zugänglich ist; und, vor Beendigung des Erzeugungsschritts,
Durchführen eines Verifikationsschritts (**33**) zum Verifizieren der Integrität der von dem Erzeugungsschritt benötigten Datei,
wobei der Verifikationsschritt einen Reparaturschritt

(**38**) zum Reparieren jeder der benötigten Dateien aufweist, deren Integrität nicht verifiziert werden kann, indem die Inhalte der Datei unter Verwendung einer Vorlagekopie der Datei ersetzt werden, die in einem zweiten Speicher (**10**) gespeichert ist, auf den der Prozessor nur Lesezugriff hat.

2. Verfahren nach Anspruch 1, wobei die eingehende Anfrage durch einen Filter (**7**) aufgefangen (**31**) wird, der die Anfrage nach Beendigung des Verifikationsschritts freigibt.

3. Verfahren nach einem der vorstehenden Ansprüche, wobei in dem Verifikationsschritt ein Sicherheitswert berechnet (**43**) wird, indem eine Verifikationsfunktion auf die Inhalte der Datei angewendet wird, und der Sicherheitswert mit einem zugehörigen Referenzwert verglichen (**44**) wird.

4. Verfahren nach Anspruch 3, wobei der Referenzwert bezüglich der Datei bestimmt wird, indem eine entsprechende Signatur entschlüsselt (**42**) wird, die in dem ersten Speicher gespeichert ist.

5. Verfahren nach Anspruch 4, wobei die Signatur aus einem Kopfabschnitt (**62**, **65**, **68**) der Datei gelesen (**41**) wird.

6. Verfahren nach Anspruch 4, wobei die Signatur von einer Nachschlagetabelle (**690**) gelesen wird, die Dateikennungen (**691**) mit zugehörigen Signaturen (**692**) in Beziehung setzt.

7. Verfahren nach einem der vorstehenden Ansprüche, wobei in dem Reparaturschritt die Inhalte der Vorlagedatei von dem zweiten Speicher gelesen (**37**) und in die entsprechende Datei in den ersten Speicher geschrieben (**38**) werden.

8. Verfahren nach Anspruch 7, wobei in dem Reparaturschritt ferner die Signatur der Datei in dem zweiten Speicher gelesen und in den ersten Speicher geschrieben wird.

9. Verfahren nach einem der vorstehenden Ansprüche, wobei der zweite Speicher ein nicht wiederbeschreibbares Aufzeichnungsmedium aufweist.

10. Verfahren nach Anspruch 9, wobei das Aufzeichnungsmedium eine CD-ROM ist.

11. Verfahren nach einem der vorstehenden Ansprüche, ferner mit einem Initialisierungsschritt, in dem die Dateien in dem ersten und dem zweiten Speicher gespeichert werden, wobei der Initialisierungsschritt für jede Datei die folgenden Schritte umfaßt:
Anwenden einer Verifikationsfunktion auf die Inhalte der Datei, um einen zugehörigen Referenzwert zu erzeugen (**51**),
Verschlüsseln (**53**) des Referenzwerts, um eine Signatur (**62**) zu erhalten, und
Speichern (**56**) der Dateiinhalte und der Signatur in sowohl dem ersten als auch dem zweiten Speicher.

12. Verfahren nach Anspruch 11, wobei in dem Verschlüsselungsschritt ein erster Schlüssel (**19**) verwendet wird, der von einem tragbaren Schlüsselträger (**16**) gelesen wird, der während des Verschlüsselungsvorgangs einer Schlüsselzugangsvorrichtung (**17**) präsentiert und anschließend davon entfernt wird.

13. Verfahren nach Anspruch 12, wobei in dem Verifikationsschritt die gespeicherte Signatur entschlüsselt wird, indem ein zweiter Schlüssel (**82**) verwendet wird, der für den Prozessor verfügbar ist.

14. Verfahren nach einem der vorstehenden Ansprüche, wobei ein Alarmzustand erzeugt (**36**) wird, falls in dem Verifikationsschritt bestimmt wird, daß die Integrität einer Datei nicht verifiziert werden kann.

15. Verfahren nach einem der vorstehenden Ansprüche, wobei die beantragte Information in Form einer Web-Seite vorliegt.

16. Verfahren nach einem der vorstehenden Ansprüche, wobei die Anfrage über das Internet (**3**) empfangen wird.

17. Gerät zum Beantworten einer Informationsanfrage, mit wenigstens einem Prozessor (**20**),
einem ersten Speicher (**9**), der für den Prozessor zum Lesen und Schreiben zugänglich ist,
einem zweiten Speicher (**10**), der für den Prozessor nur zum Lesen zugänglich ist,
wobei der wenigstens eine Prozessor dazu betreibbar ist, folgendes bereitzustellen:
eine Erzeugungseinrichtung (**6**) zum Erzeugen geforderter Information unter Verwendung der Inhalte wenigstens einer Datei (**60**, **63**, **66**, **69**), die in dem ersten Speicher gespeichert ist, und
eine Verifikationseinrichtung (**11**), um vor dem Betrieb der Erzeugungseinrichtung die Integrität jeder von der Erzeugungseinrichtung benötigten Datei zu verifizieren, und
ferner mit einer Reparatureinrichtung (**12**), um jede der benötigten Dateien, deren Integrität nicht verifiziert werden kann, zu reparieren, indem die Inhalte der Datei unter Verwendung einer in dem zweiten Speicher gespeicherten Vorlagekopie der Datei ersetzt werden.

18. Gerät nach Anspruch 17 mit einem Filter (**7**) zum Auffangen der eingehenden Anfrage und Freigeben der Anfrage nach Beendigung der Dateiverifikation.

19. Gerät nach Anspruch 17 oder 18, wobei die

Verifikationseinrichtung eine Berechnungseinrichtung zum Berechnen eines Sicherheitswerts (**81**) durch Anwenden einer Verifikationsfunktion (**80**) auf die Inhalte der Datei und eine Vergleichseinrichtung (**84**) zum Vergleichen des Sicherheitswerts mit einem zugehörigen Referenzwert (**83**) aufweist.

20. Gerät nach Anspruch 19 mit einer Entschlüsselungseinrichtung (**72**) zum Entschlüsseln einer in dem ersten Speicher gespeicherten Signatur (**62**), um den Referenzwert zu bestimmen.

21. Gerät nach Anspruch 20 mit einer Leseeinrichtung zum Lesen der Signatur aus einem Kopfabschnitt (**62**, **65**, **68**) der Datei.

22. Gerät nach Anspruch 20 mit einer Leseeinrichtung zum Lesen der Signatur aus einer Nachschlagetabelle (**690**), die Dateikennungen (**691**) mit zugehörigen Signaturen (**692**) in Beziehung setzt.

23. Gerät nach einem der Ansprüche 17 bis 22, wobei die Reparatureinrichtung eine Einrichtung zum Lesen der Inhalte der Vorlagedatei von dem zweiten Speicher und eine Einrichtung zum Schreiben der Inhalte in die entsprechende Datei in dem ersten Speicher aufweist.

24. Gerät nach Anspruch 23, wobei die Reparatureinrichtung ferner eine Einrichtung zum Lesen der Signatur der Datei in dem zweiten Speicher und Schreiben der Signatur in den ersten Speicher aufweist.

25. Gerät nach einem der Ansprüche 17 bis 24, wobei der zweite Speicher ein nicht wiederbeschreibbares Aufzeichnungsmedium aufweist.

26. Gerät nach Anspruch 25, wobei das Aufzeichnungsmedium eine CD-ROM ist.

27. Gerät nach einem der Ansprüche 17 bis 26, mit einer Initialisiereinrichtung zum Speichern der Dateien in dem ersten und dem zweiten Speicher, wobei die Initialisiereinrichtung aufweist:
eine Einrichtung zum Anwenden einer Verifikation auf die Inhalte jeder Datei, um einen zugehörigen Referenzwert zu erzeugen,
eine Verschlüsselungseinrichtung (**15**) zum Verschlüsseln des Referenzwerts, um eine Signatur zu erhalten, und
eine Einrichtung (**14**) zum Speichern der Dateiinhalte und der Signatur sowohl in dem ersten als auch in dem zweiten Speicher.

28. Gerät nach Anspruch 27, wobei die Verschlüsselungseinrichtung dazu betreibbar ist, einen ersten Schlüssel (**19**) zu verwenden, der aus einem tragbaren Schlüsselträger (**16**) gelesen wird, der während des Verschlüsselungsvorgangs in Gebrauch einer Schlüsselzugriffsvorrichtung (**17**) des Geräts präsentiert und anschließend davon entfernt wird.

29. Gerät nach Anspruch 28, wobei die Verifikationseinrichtung eine Entschlüsselungseinrichtung (**72**) zum Entschlüsseln der gespeicherten Signatur unter Verwendung eines zweiten Schlüssels aufweist, der für den Prozessor zugänglich ist.

30. Gerät nach einem der Ansprüche 17 bis 29 mit einer Alarmeinrichtung (**13**) zum Erzeugen eines Alarmzustands, falls die Verifikationseinrichtung bestimmt, daß die Integrität einer Datei nicht verifiziert werden kann.

31. Gerät nach einem der Ansprüche 17 bis 30, wobei die benötigte Information in Form einer Web-Seite vorliegt.

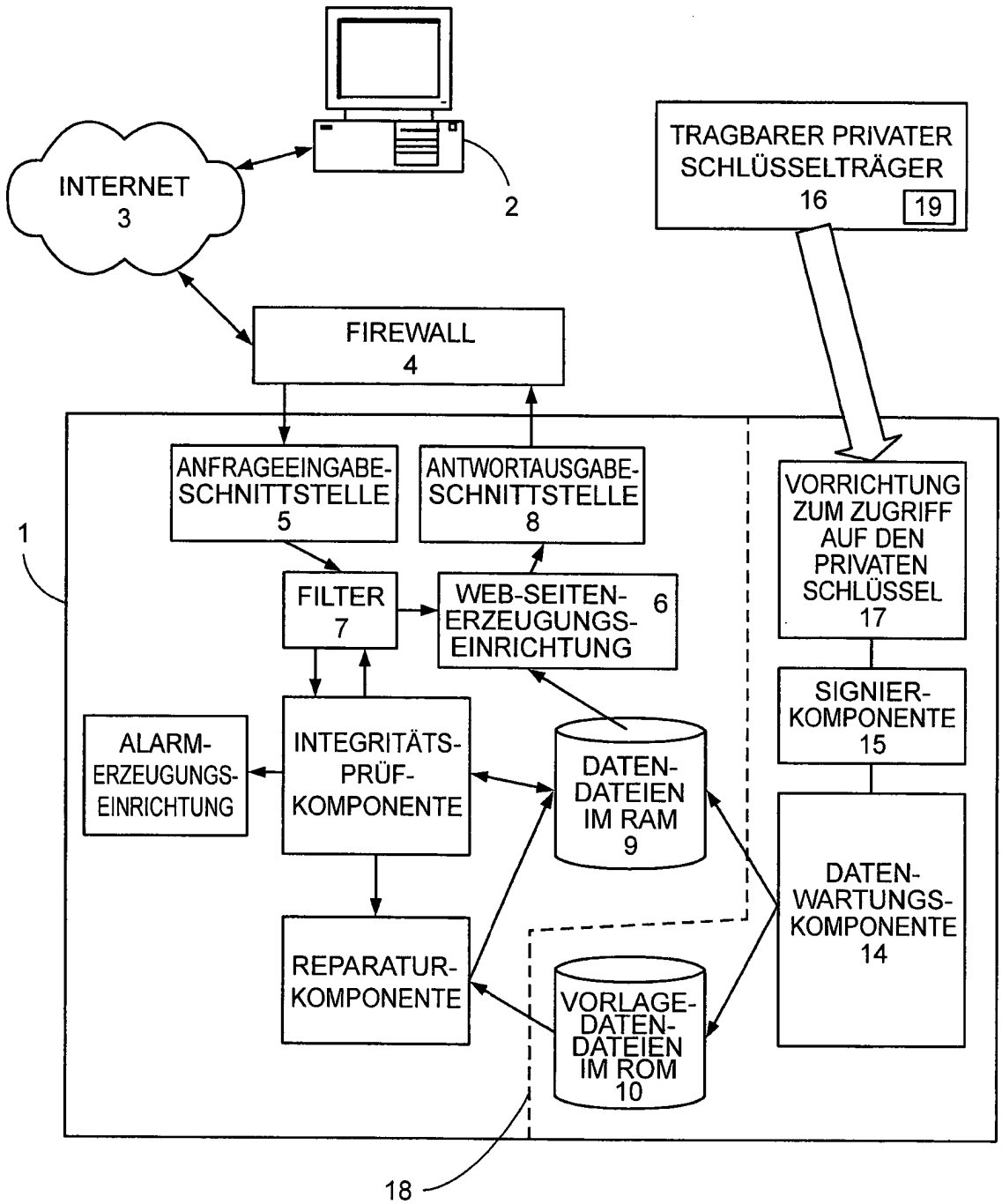
32. Gerät nach einem der Ansprüche 17 bis 31, wobei das Gerät in einem Server (**1**) besteht, der im Gebrauch an das Internet (**3**) angeschlossen ist, um die Informationsanfragen zu empfangen.

33. Speichermedium, das Befehle speichert, die bei Ausführung durch einen Prozessor den Prozessor dazu veranlassen, ein Verfahren gemäß einem der Ansprüche 1 bis 16 auszuführen.

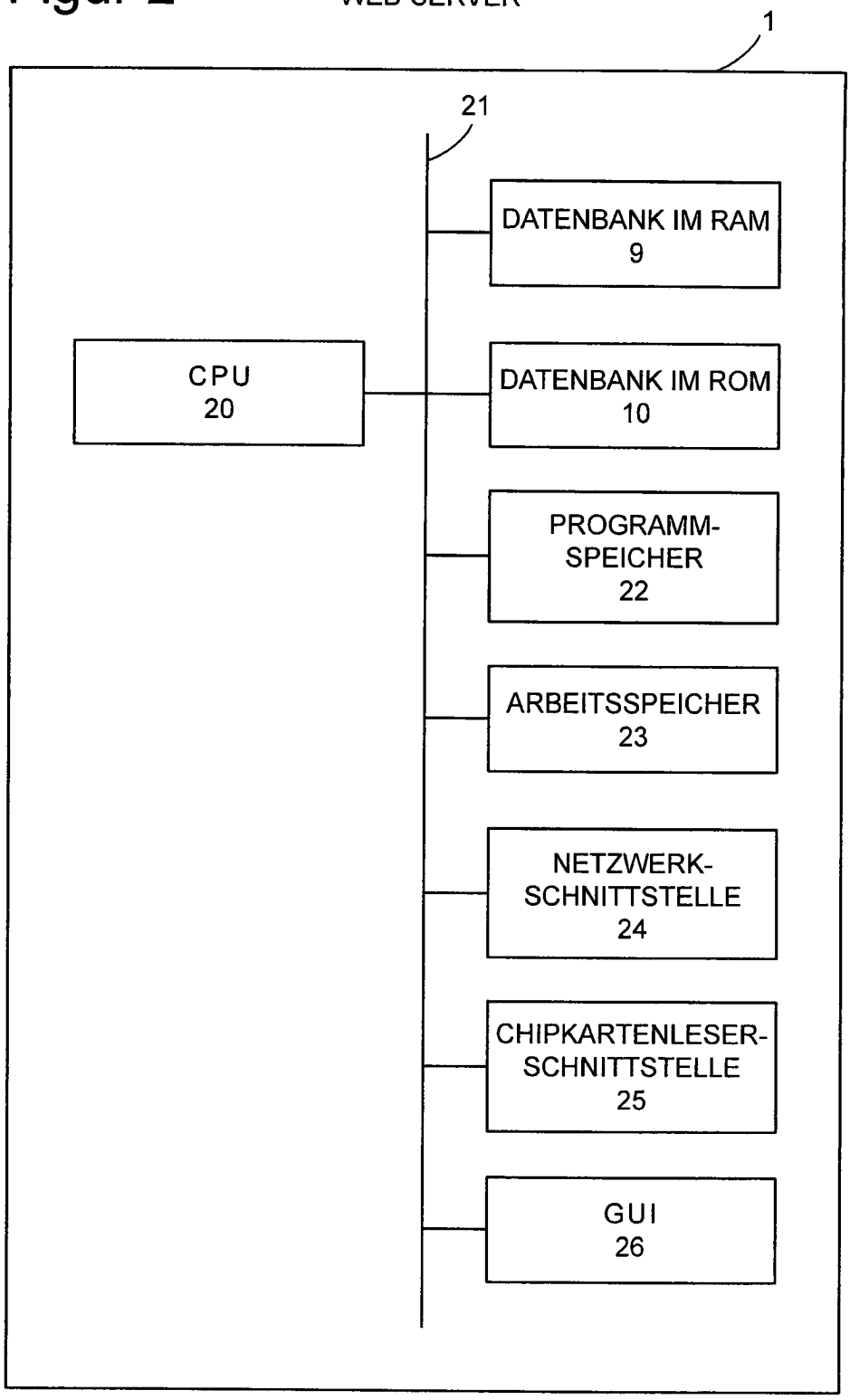
Es folgen 9 Blatt Zeichnungen

Anhängende Zeichnungen

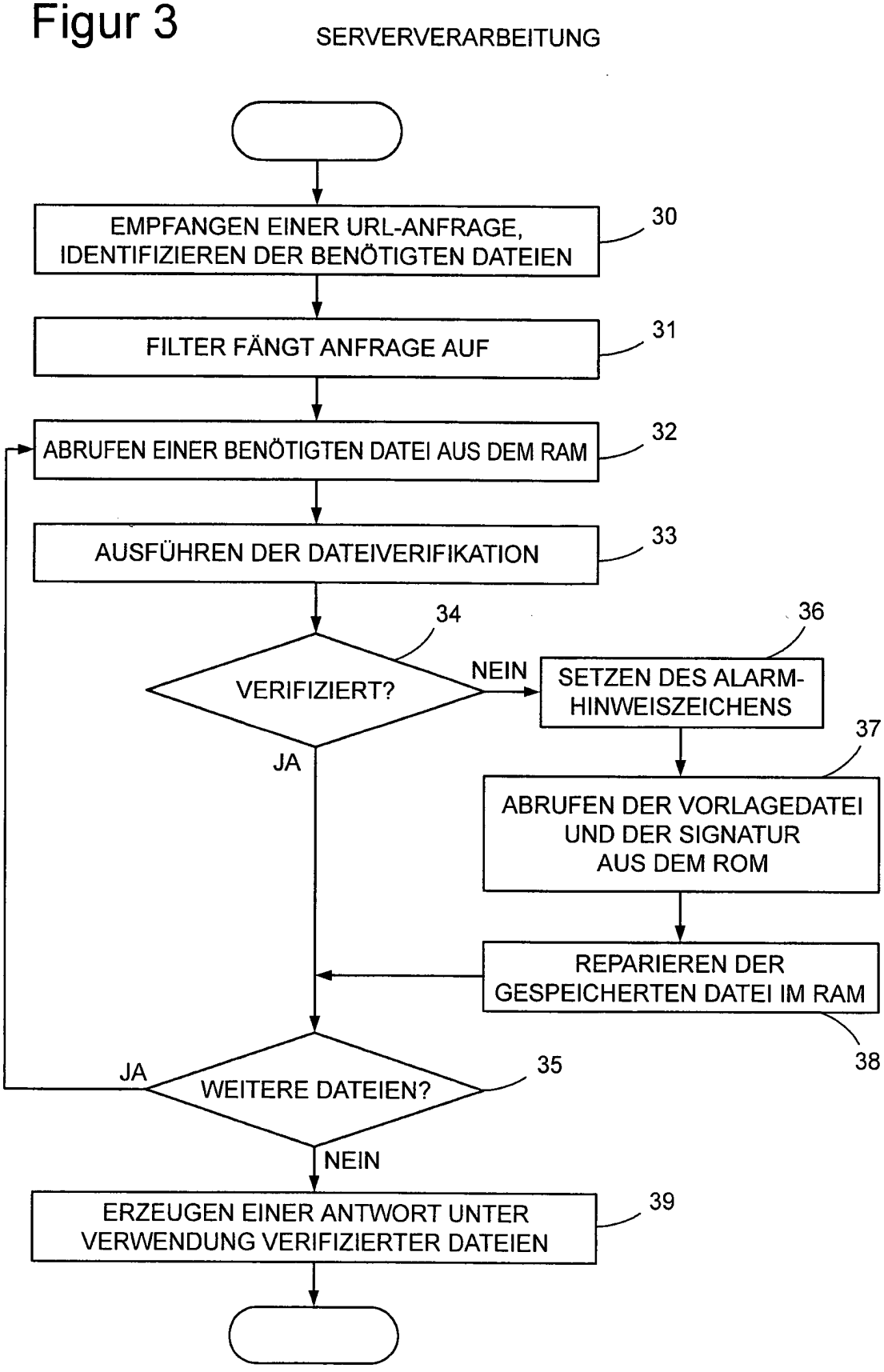
Figur 1



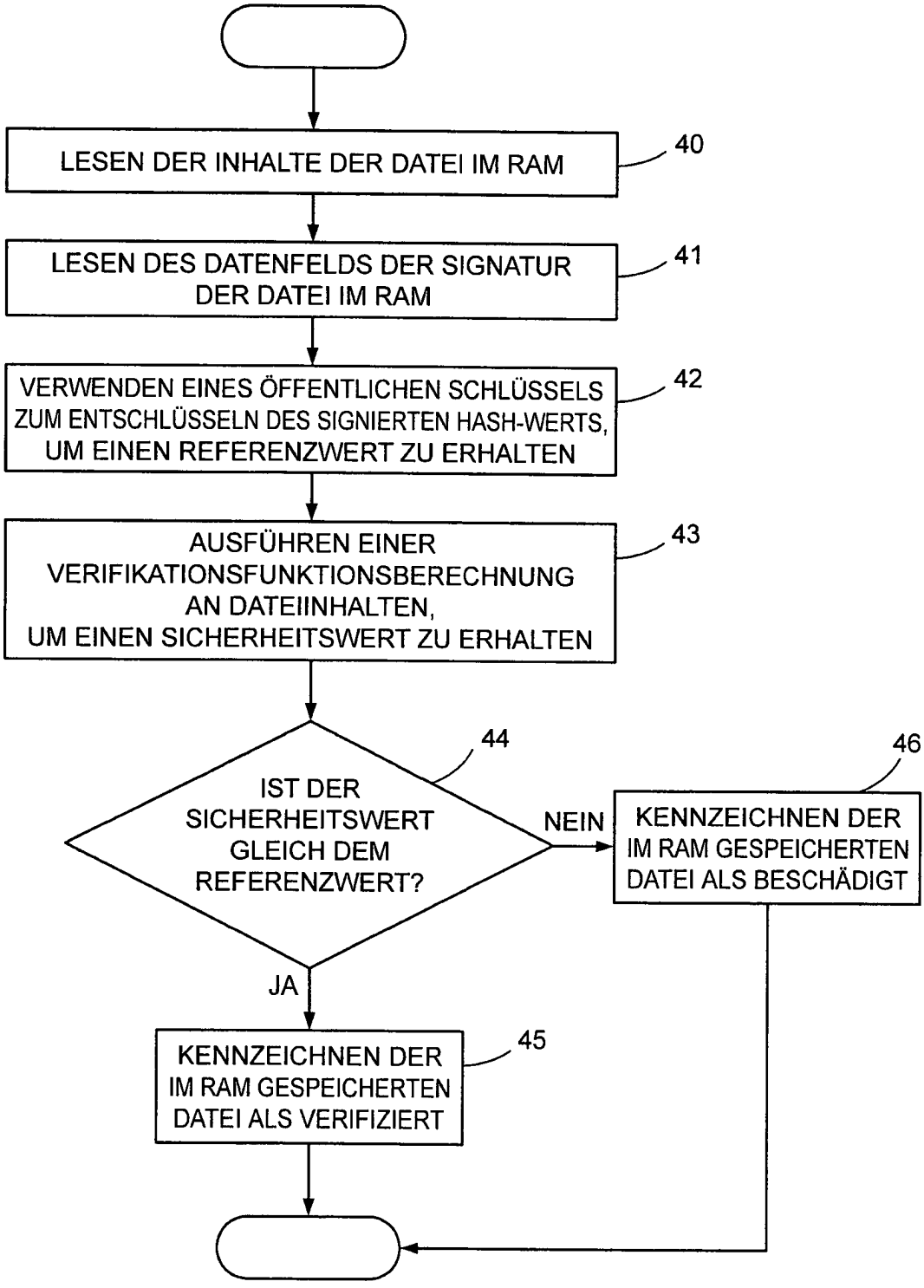
Figur 2


WEB-SERVER
1
21
DATENBANK IM RAM
9
CPU
20
DATENBANK IM ROM
10
PROGRAMM-
SPEICHER
22
ARBEITSSPEICHER
23
NETZWERK-
SCHNITTSTELLE
24
CHIPKARTENLESER-
SCHNITTSTELLE
25
GUI
26

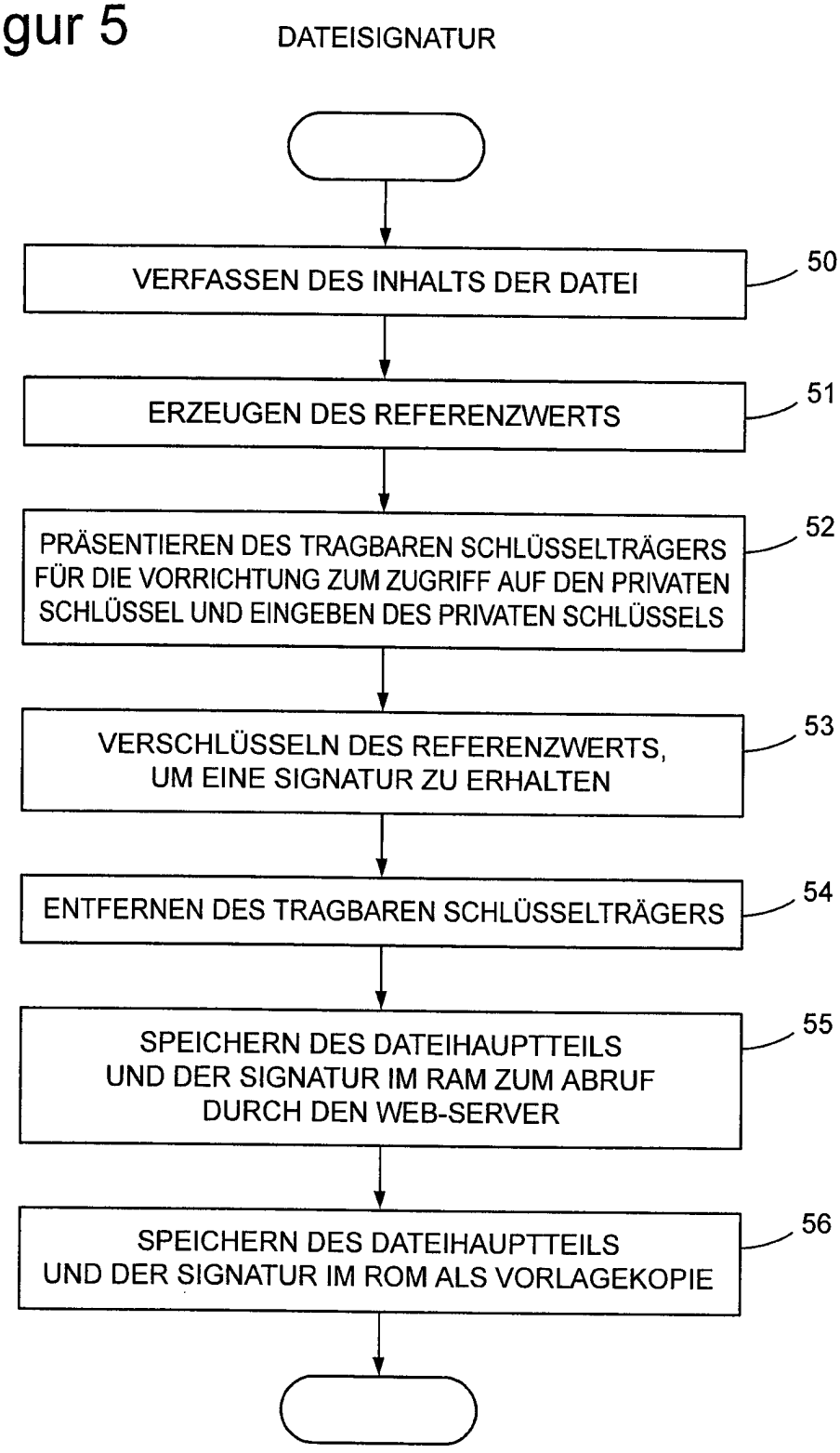
Figur 3


SERVERVERARBEITUNG
EMPFANGEN EINER URL-ANFRAGE, IDENTIFIZIEREN DER BENÖTIGTEN DATEIEN
30
FILTER FÄNGT ANFRAGE AUF
31
ABRUFEN EINER BENÖTIGTEN DATEI AUS DEM RAM
32
AUSFÜHREN DER DATEIVERIFIKATION
33
34
VERIFIZIERT?
NEIN
36
SETZEN DES ALARM-HINWEISZEICHENS
37
JA
ABRUFEN DER VORLAGEDATEI UND DER SIGNATUR AUS DEM ROM
REPARIEREN DER GESPEICHERTEN DATEI IM RAM
38
JA
WEITERE DATEIEN?
35
NEIN
ERZEUGEN EINER ANTWORT UNTER VERWENDUNG VERIFIZIERTER DATEIEN
39

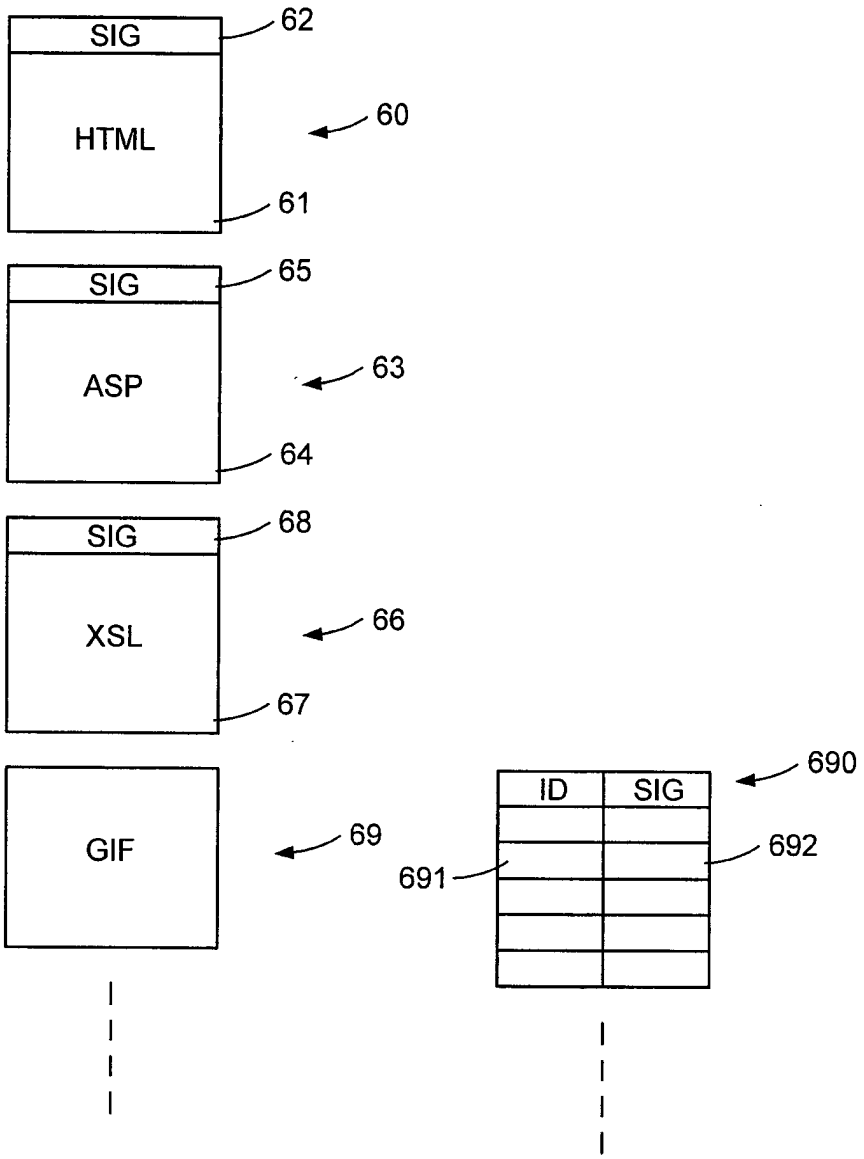
Figur 4 DATEIVERIFIKATION



Figur 5

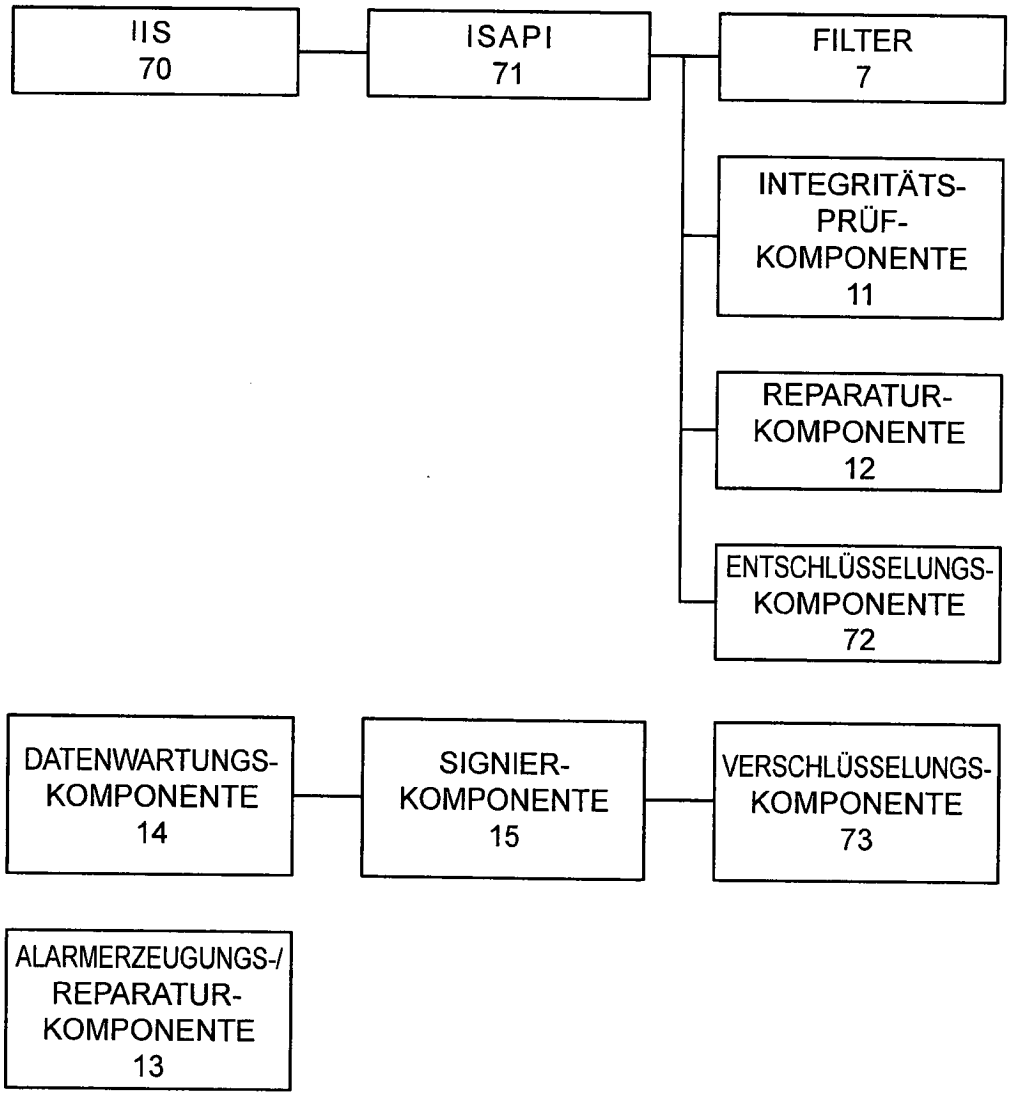

DATEISIGNATUR
VERFASSEN DES INHALTS DER DATEI
50
ERZEUGEN DES REFERENZWERTS
51
PRÄSENTIEREN DES TRAGBAREN SCHLÜSSELTRÄGERS FÜR DIE VORRICHTUNG ZUM ZUGRIFF AUF DEN PRIVATEN SCHLÜSSEL UND EINGEBEN DES PRIVATEN SCHLÜSSELS
52
VERSCHLÜSSELN DES REFERENZWERTS, UM EINE SIGNATUR ZU ERHALTEN
53
ENTFERNEN DES TRAGBAREN SCHLÜSSELTRÄGERS
54
SPEICHERN DES DATEIHAUPTTEILS UND DER SIGNATUR IM RAM ZUM ABRUF DURCH DEN WEB-SERVER
55
SPEICHERN DES DATEIHAUPTTEILS UND DER SIGNATUR IM ROM ALS VORLAGEKOPIE
56

Figur 6 DATENSTRUKTUR



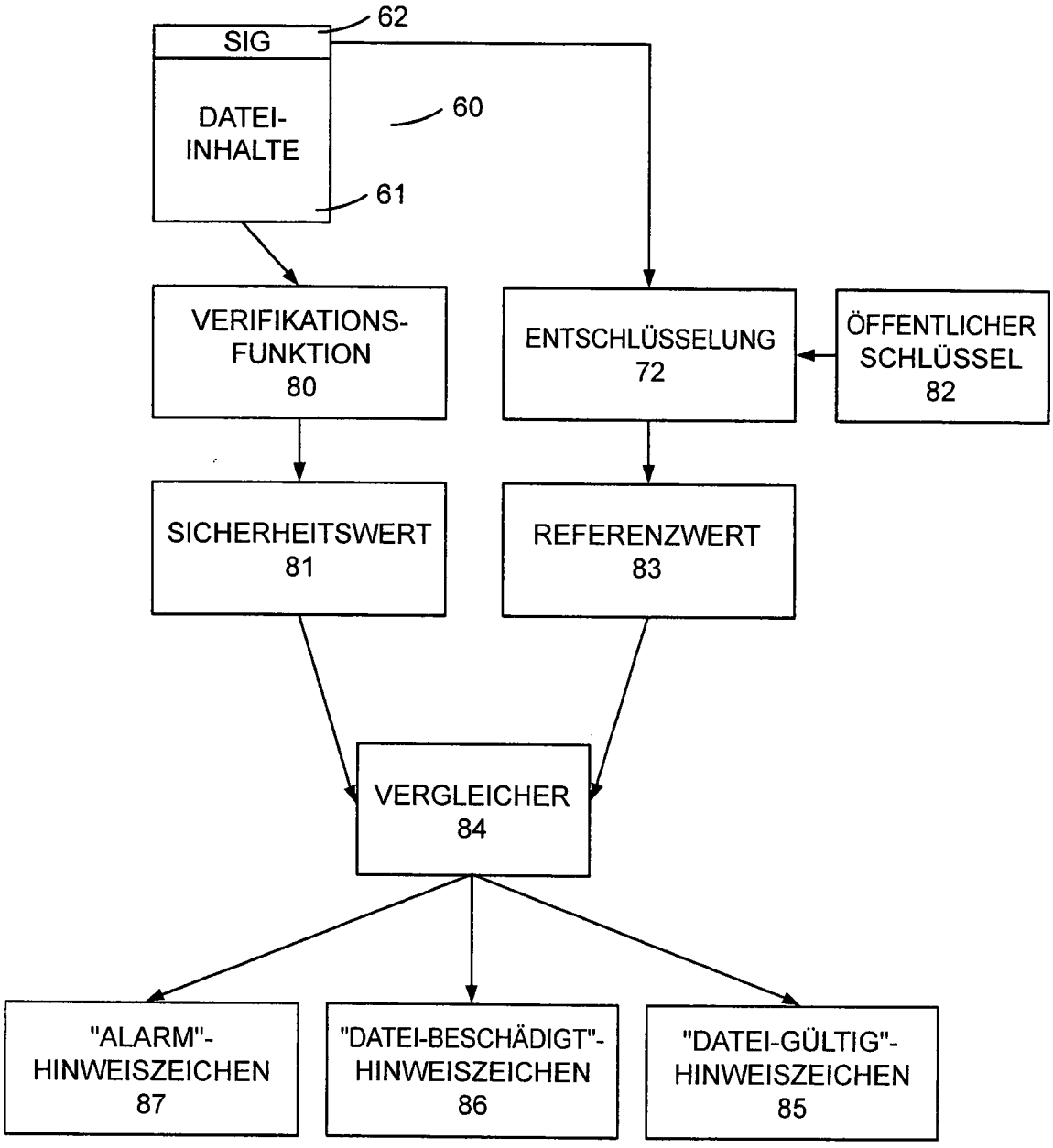
Figur 7

DEM WEB-SERVER BEREITGESTELLTE SOFTWARE



Figur 8

DATENFLUSS WÄHREND DER VERIFIKATION



Figur 9

BEREITSTELLEN VON PROGRAMMEN FÜR DEN SERVER