



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 600 36 112 T2 2007.12.06**

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 197 032 B1**

(21) Deutsches Aktenzeichen: **600 36 112.8**

(86) PCT-Aktenzeichen: **PCT/US00/18546**

(96) Europäisches Aktenzeichen: **00 948 590.5**

(87) PCT-Veröffentlichungs-Nr.: **WO 2001/001627**

(86) PCT-Anmeldetag: **29.06.2000**

(87) Veröffentlichungstag
der PCT-Anmeldung: **04.01.2001**

(97) Erstveröffentlichung durch das EPA: **17.04.2002**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **22.08.2007**

(47) Veröffentlichungstag im Patentblatt: **06.12.2007**

(51) Int Cl.⁸: **H04L 9/08 (2006.01)**
H04L 9/32 (2006.01)

(30) Unionspriorität:

141571 P	29.06.1999	US
167453 P	23.11.1999	US
188834 P	10.03.2000	US
574687	17.05.2000	US

(73) Patentinhaber:

VeriSign, Inc., Mountain View, Calif., US

(74) Vertreter:

Strehl, Schübel-Hopf & Partner, 80538 München

(84) Benannte Vertragsstaaten:

**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE**

(72) Erfinder:

FORD, Warwick S., Cambridge, MA 02138, US

(54) Bezeichnung: **SERVERUNTERSTÜTZTE WIEDERGEWINNUNG EINES STARKEN GEHEIMNISSES AUS EINEM SCHWACHEN GEHEIMNIS**

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

1. Technisches Gebiet

[0001] Die Erfindung betrifft allgemein die sichere Wiedergewinnung stark geheimer Daten eines Benutzers, wenn der Benutzer entsprechend schwach geheime Daten liefert, wie ein von ihm gewähltes Passwort. Beispielsweise können bei Computernetzwerkanwendungen die stark geheimen Daten ein Verschlüsselungsschlüssel sein, der dazu verwendet wird, die hoch sensitiven privaten Daten des Benutzers (wie den bei Kryptografie mit einem öffentlichen Schlüssel verwendeten Privatschlüssel des Benutzers) zu schützen. Bei diesem Beispiel betrifft die Erfindung die sichere Wiedergewinnung des Verschlüsselungsschlüssels (und die sichere Wiedergewinnung des Privatschlüssels des Benutzers), wenn der Benutzer sein Passwort liefert. Gemäß einem anderen Beispiel können die stark geheimen Daten vom Benutzer dazu verwendet werden, einen Server dadurch zu authentifizieren, dass dem Server die Fähigkeit des Benutzers gezeigt wird, die stark geheimen Daten wiederzugewinnen, ohne dass der Server Daten speichern müsste, die es ermöglichen, die schwach geheimen Daten durch erschöpfende Versuche zu attackieren.

2. Hintergrundbildende Technik

[0002] Als Ergebnis der andauernden Entwicklung neuer Technologien, insbesondere auf den Gebieten von Computernetzwerken und Kommunikationsabläufen, wird die Verwendung großer Computernetzwerke wie des Internets immer weiter verbreitet. Dies hat zu einer Zunahme des elektronischen Handels und anderer elektronischer Transaktionen geführt, die über diese Netzwerke ausgeführt werden. Es hat auch zu erhöhter Flexibilität der Benutzer geführt, da diese zunehmend von einer beliebigen Anzahl von Orten und/oder Vorrichtungen auf diese Netzwerke zugreifen können. Die Zunahme der elektronischen Transaktionen hat zu einem entsprechend erhöhten Bedarf an Sicherheit für diese Transaktion geführt; jedoch legt die erhöhte Flexibilität der Sicherheit zusätzliche Erfordernisse auf, da alle Sicherheitsmaßnahmen vorzugsweise Benutzern auch dann dienen sollen, wenn sie Roamingabläufe über das Netzwerk hinweg ausführen.

[0003] Bei einem üblichen Szenario kann der Benutzer von vielen verschiedenen Orten auf das Computernetzwerk zugreifen, und er kann es wünschen, seinen Privatschlüssel von jedem Ort aus zu verwenden. Jedoch muss er an jedem Ort über eine Vorrichtung (nachfolgend als "Benutzerterminal" bezeichnet), die keinerlei Daten für den Benutzer speichern kann oder nicht speichert, außer in vorübergehenden Perioden, auf das Netzwerk zugreifen. Beispielsweise kann ein Angestellter von verschiedenen Termi-

nals in Gebäuden der Firma auf das Zentralcomputernetzwerk einer Firma zugreifen, oder ein Verbraucher kann von jedem beliebigen Webbrowser aus auf das Internet zugreifen, oder er kann von einer Kundentelefonzelle auf ein privates Netzwerk zugreifen. Dem Benutzerterminal kann typischerweise dahingehend getraut werden, dass es seinen Code auf vertrauenswürdige Weise ausführt, um die Geheimhaltung sensibler Daten (Beispielsweise den Privatschlüssel des Benutzers oder Geheimdaten, die gemeinsam mit einem Anwendungsserver verwendet werden) während der Periode aufrecht zu erhalten, in der der Benutzer aktiv dieses Terminal nutzt, und um sensitive Daten sicher zu zerstören, wenn der Benutzer ihre Nutzung beendet hat. Demgemäß könnten die Privatdaten des Benutzers am Benutzerterminal sicher genutzt werden, wenn dieses mit gewisser Sicherheit eine Kopie dieser Privatdaten erhalten könnte.

[0004] Bei einer Vorgehensweise werden die Privatdaten in irgendeinem sicheren Hardwarespeicher oder einem Verarbeitungschip, wie einer Smartcard, gespeichert. Der Hardwarechip wird körperlich mit dem Benutzerterminal verbunden, und die Privatdaten werden dem Benutzer zugänglich gemacht. Diese Vorgehensweise leidet unter hohen Kosten und Unzweckdienlichkeit für den Benutzer, da speziell Hardware erforderlich ist, was dies für viele Anwendungen ungeeignet macht.

[0005] Bei einer anderen Vorgehensweise werden die Privatdaten mit Unterstützung anderer mit dem Netzwerk verbundener Vorrichtungen (nachfolgend als "Server" bezeichnet) wieder gewonnen. Bei einem Beispiel bildet die Wiedergewinnung der Privatdaten einen Teil des Sitzungsloginprozesses durch den Benutzer. Der Benutzer zeigt seine Berechtigung durch Angeben eines Benutzerkontennamens und eines Passworts, die mäßigen (jedoch nicht extremen) Erratekontrollen unterzogen werden. Insbesondere wird jede Partei, die einen Passworterrateangriff versucht, auf eine kleine Anzahl von Versuchen eingeschränkt, und anhand dieser Kontrolle ist es Benutzern möglich, vernünftig freundliche Passwörter zu wählen. Wenn der Benutzer einmal authentifiziert ist, gewinnt das Benutzerterminal die Privatdaten des Benutzers mit Unterstützung der Server wieder.

[0006] Das Problem der Wiedergewinnung von Privatdaten, wie eines Verschlüsselungsschlüssels, in einem staatenlosen Benutzerterminal wurde bei früheren Arbeiten durch die Verwendung eines Servers berücksichtigt, der Geheimdaten für den Benutzer speichert und den Wiedergewinnungsprozess erleichtert. Verschiedene Protokolle unter Verwendung derartiger Server mit verschiedenen Sicherheits- und Funktionseigenschaften wurden von R. Perlman und C. Kaufman im "Secure Password-Based Protocol for Downloading a Private Key", Proc. 1999 Network

and Distributed System Security Symposium, Internet Society, Januar 1999 durchgesehen. Die in dieser Arbeit beschriebenen Protokolle sind hauptsächlich Varianten oder Ableitungen des EKE-Protokolls von Bellovin und Merritt (siehe beispielsweise S. Bellovin und M. Merritt in "Encrypted Key Exchange: Password-based protocols secure against dictionary attacks", Proc. IEEE Symposium on Research in Security and Privacy, Mai 1992; und S. Bellovin und M. Merritt in "Augmented Encrypted Key Exchange: a password-based protocol secure against dictionary attacks and password file compromise", ATT Labs Technical Report, 1994) und des SPEKE-Protokolls von Jablon (siehe beispielsweise D. Jablon in "Strong password-only authenticated key exchange", ACM Computer Communications Review, Oktober 1996; und D. Jablon in "Extended Password Protocols Immune to Dictionary Attack", Proc. of the WETICE '97 Enterprise Security Workshop, Juni 1997). Zu einschlägigen Patenten gehören das US-Patent Nr. 5,241,599 ("Cryptographic protocol for secure communications" von Bellovin und Merritt) und das US-Patent Nr. 5,440,635 ("Cryptographic protocol for remote authentication" von Bellovin und Merritt). Andere einschlägige, serverunterstützte Wiedergewinnungsprotokolle für Geheimdaten wurden von den folgenden vorgeschlagen: Gong et al. (siehe beispielsweise L. Gong, T.M.A. Lomas, R.M. Needham und J.H. Salzer, "Protecting Poorly Chosen Secrets from Guessing Attacks", IEEE Journal on Selected Areas in Communications, Vol. 11, Nr. 5, Juni 1993, S. 648-656; L. Gong, "Optimal Authentication Protocols Resistant to Password Guessing Attacks", Proc. 8th IEEE Computer Security Foundations Workshop, Irland, 13. Juni 1995, S. 24-29; und L. Gong "Increasing Availability and Security of an Authentication Service", IEEE Journal on Selected Areas in Communications, Vol. 11, Nr. 5, Juni 1993, S. 657-662); Wu (siehe beispielsweise T. Wu in "The Secure Remote Password Protocol", Proc. 1998 Network and Distributed System Security Symposium, Internet Society, Januar 1998, S. 97-111); und Halevi und Krawczyk (siehe beispielsweise S. Halevi und H. Krawczyk in "Public-key cryptography and password protocols", Proceedings of the Fifth ACM Conference on Computer and Communications Security, 1998).

[0007] Jedoch leiden alle diese obigen Verfahren unter einem deutlichen Mangel. Der Server bildet eine wesentliche verletzungsanfällige Stelle. Wenn es ein Serverbetreiber, oder irgendjemand, der einem Server schadet, wünscht, ein Passwort eines Benutzers oder Privatdaten (jedes derselben ermöglicht es dem Angreifer im Wesentlichen, sich als Benutzer auszugeben) zu bestimmen, sind funktionsfähige Attacken möglich, trotz Gesichtspunkten einiger Vorgehensweisen, die die Sensitivität der im Server gespeicherten Daten minimieren. Beispielsweise speichert der Server, gemäß bestimmten der o.a. Ar-

beiten, nicht das Passwort des Benutzers, sondern stattdessen speichert er einen Wert, der als unidirektionale Funktion des Passworts berechnet wurde. Jedermann, der diesen Wert erfährt, kann dazu in der Lage sein, das Passwort dadurch zu bestimmen, dass er die unidirektionale Funktion erschöpfend auf geratene Passwörter anwendet und das Ergebnis mit dem gespeicherten Wert vergleicht. Allgemein gesagt, leiden die zuvor angegebenen Vorgehensweisen unter der Schwäche, dass jeder, der auf die Serverdatenbank zugreifen kann oder irgendwelche Drosselungs- oder Aussperrmechanismen im Server deaktivieren kann, Passwörter erschöpfend ausprobieren kann, bis er in ein Konto des Benutzers im Server eingedrungen ist.

[0008] Bei einigen Anwendungsszenarien kann die obige Schwäche die Attraktivität der serverunterstützten Vorgehensweise zum Wiedergewinnen von Privatdaten deutlich unterminieren. Beispielsweise stört das obige Angriffsszenario deutlich die Nichtverweigerungseigenschaften, wie sie ansonsten der digitalen Signaturtechnologie eigen sind. Wenn ein sich im Netzwerk bewegendes Benutzer unter Verwendung seines Verschlüsselungsschlüssels von einem Benutzerterminal aus eine Transaktion digital unterzeichnet und er es später wünscht, seine digitale Signatur abzustreiten, kann er auf plausible Weise den Anspruch erheben, dass der Serverbetreiber oder jemand, der den Server gestört hat, seinen Privatschlüssel auf die oben beschriebene Weise erhalten hat und die Transaktion digital unterzeichnet hat, wobei er sich als Benutzer ausgegeben hat. Die Gefahren und die Haftung, denen sich ein Serverbetreiber gegenüber sieht, sind verringert, wenn er auf rechtfertigbare Weise Ansprüchen von Benutzern gegenüber treten kann, er, oder sein Personal hätte sich als Benutzer ausgegeben.

[0009] So besteht Bedarf an einer Vorgehensweise, die es einem Benutzerterminal ermöglicht, Privatdaten eines Benutzers mit Unterstützung von Servern wiederzugewinnen, während Resistenz gegen Attacken auf die Server erhalten bleibt. Allgemeiner, kann das Problem der Wiedergewinnung von Privatdaten bei einem staatenlosen Client auf das Problem reduziert werden, stark geheime Daten für einen Benutzer aus schwach geheimen Daten des Benutzers, wie einem Passwort, zu erzeugen und wieder zu gewinnen. Die stark geheimen Daten können in einem symmetrischen Konvergenzschichtsystem als Verschlüsselungsschlüssel verwendet werden, um eine beliebige Menge von Privatdaten, die in verschlüsselter Form in einem in weitem Umfang zugänglichen Speicherort vorhanden sein können, zu verschlüsseln und wiederzugewinnen.

[0010] Es besteht auch Bedarf an Vorgehensweisen, die es einem Benutzer erlauben, einen Anwendungsserver von einem staatenlosen Benutzertermi-

nal aus auf Grundlage eines angegebenen Passworts zu authentifizieren. Zu aktuellen Vorgehensweisen, wie dem Kerberos-Authentifizierungsverfahren (siehe beispielsweise J.T. Kohl und B.C. Neuman in "The Kerberos Network Authentication Service (V5), Request for Comments (RFC) 1510, Internet Activities Board, 1993), gehört es, dass sich der Benutzer zuerst an einem Authentifizierungsserver und anschließend an einem Anwendungsserver unter Verwendung eines kryptografischen "Berechtigungsscheins" vom Authentifizierungsserver als berechtigt ausweist. Jedoch leiden diese Vorgehensweisen unter dem Mangel, dass entweder der Anwendungsserver oder der Authentifizierungsserver Daten enthält, die es einem Angreifer (entweder innerhalb oder außerhalb der Organisation, die den Server betreibt), wenn sie für diesen frei liegen, erlauben, Passwörter erschöpfend zu erraten und wahrscheinlich das Passwort des Benutzers zu bestimmen. Diese Probleme können dann vermieden werden, wenn die Benutzerauthentifizierung darauf beruht, dass ein Benutzer dem Anwendungsserver oder dem Authentifizierungsserver Augenscheinlichkeit dafür liefert, dass er stark geheime Daten statt schwach geheimer Daten kennt. Es besteht auch Bedarf an Vorgehensweisen, die es einem Benutzer ermöglichen, eine digitale Signatur ausgehend von einem staatenlosen Benutzerterminal, in das ein Passwort eingegeben wird, zu erzeugen. Eine Vorgehensweise, diese Forderung zu erfüllen, besteht darin, den Privatschlüssel des Benutzers im Terminal auf die oben skizzierte Weise wiederzugewinnen und die digitale Signatur im Benutzerterminal zu berechnen. Zu einer anderen Vorgehensweise zum Erfüllen der Forderung, bei der es nicht erforderlich ist, den Privatschlüssel an einem Ort zusammenzusetzen, gehören Kommunikationsvorgänge mit mehreren Servern, von denen jeder einen unabhängigen Teil des Privatschlüssels des Unterzeichnenden Benutzers speichert. Derartige Server erzeugen jeweils einen Teil der digitalen Signatur, und die Teile werden im Benutzerterminal kombiniert, um die vollständige digitale Signatur zu liefern. Während andere relevante Arbeiten an derartigen Verfahren das Ziel erreichen, den Privatschlüssel nicht an einer Stelle zusammenzusetzen, leiden derartiger Arbeiten unter der Schwäche, dass ein oder mehrere Server, die am Signierprozess Teil haben, Daten speichern, was es ermöglicht, das Passwort des Benutzers erschöpfend zu attackieren. Demgemäß besteht die Gefahr, dass jeder, der einen derartigen Server stört, das Passwort des Benutzers durch erschöpfendes Erraten bestimmen kann, was es, wenn dies erfolgreich ist, dem Angreifer ermöglicht, digitale Signaturen zu fälschen, um vorzugeben, sie würden von diesen Benutzer herrühren. Das Problem kann dadurch gelindert werden, dass eine Authentifizierung bei denjenigen Servern, die Teile der digitalen Signatur erzeugen, dadurch erfolgt, dass die Kenntnis stark geheimer Daten des Benutzers demonstriert wird, die unter Umständen auf Grundlage der Angabe

schwach geheimer Daten des Benutzers wiedergewonnen wurden, anstatt einer direkten Authentifizierung auf Grundlage der schwach geheimen Daten des Benutzers selbst.

[0011] Demgemäß besteht Bedarf an einer Vorgehensweise, die es einem Benutzerterminal erlaubt, stark geheime Daten eines Benutzers mit Unterstützung durch Server aus schwach geheimen Daten wiederzugewinnen, während Resistenz gegen Attacken auf Server erhalten bleibt.

[0012] Gesichtspunkte der Erfindung sind in den beigefügten Ansprüchen dargelegt.

[0013] Gemäß einer Ausführungsform beinhaltet ein Verfahren zum Erstellen (**300, 500**) stark geheimer Daten eines Benutzers (**100**), um eine anschließende Wiedergewinnung (**400, 600**) der stark geheimen Daten zu ermöglichen, die folgenden Schritte. Es werden schwach geheime Daten, beispielsweise ein Passwort, für den Benutzer (**110**) bestimmt (**320**). Der Benutzer führt eine Authentifizierung (**310**) an Servern (**130**) aus, zu denen Geheimdaten speichernde Server gehören (vorzugsweise mindestens zwei Geheimdaten speichernde Server). Jeder Geheimdaten speichernde Server verfügt über entsprechende Servergeheimdaten. Ein Erzeugungs-Client (**120**), der möglicherweise durch die Geheimdaten speichernde Server (**130**) unterstützt wird, berechnet (**330, 530**) die stark geheimen Daten des Benutzers. Die stark geheimen Daten sind eine Funktion der schwach geheimen Daten des Benutzers und der Servergeheimdaten. Bei einer bevorzugten Ausführungsform werden für jeden Geheimdaten speichernden Server Geheimkomponenten berechnet (**534**). Jede Geheimkomponente ist eine Funktion der schwach geheimen Daten und der stark geheimen Daten für diesen Geheimdaten speichernden Server. Die Geheimkomponenten werden kombiniert (**536**), um die stark geheimen Daten zu erzeugen. Der Erzeugungs-Client (**120**) bestimmt auch (**350**) Verifikationsdaten für Verifikationsserver (**130**), vorzugsweise mindestens zwei. Die Verifikationsdaten ermöglichen es einem Verifikationsserver (**130**), zu verifizieren (**402, 602**), ob eine Vorrichtung (**220**) anschließend die stark geheimen Daten erfolgreich wiedergewonnen hat (**400, 600**). Jedoch ist der Server (**130**) rechtmäßig nicht dazu in der Lage, die schwach geheimen Daten alleine durch Zugriff auf seine Verifikationsdaten zu bestimmen. Die Verifikationsserver (**130**) können die Verifikationsdaten für anschließenden Gebrauch speichern (**355**). Der Erzeugungs-Client (**120**) kann zusätzlich die stark geheimen Daten als Kryptografieschlüssel in einem symmetrischen Kryptografiesystem verwenden, um andere Privatdaten für den Benutzer (**110**), wie den Privatschlüssel desselben, zu verschlüsseln (**370**).

[0014] Bei einer bevorzugten Ausführungsform wer-

den die stark geheimen Daten wie folgt berechnet (**330**, **530**). Der Erzeugungs-Client (**120**) berechnet Serveranfragedaten für mindestens einen der Geheimdaten speichernden Server (**130**). Die Serveranfragedaten sind eine Funktion der schwach geheimen Daten und kurzlebiger Benutzergeheimdaten, jedoch enthüllen die Serveranfragedaten keine Information zu den schwach geheimen Daten, ohne dass Kenntnis der kurzlebigen Benutzergeheimdaten vorhanden wäre. Im Ergebnis kann der Erzeugungs-Client (**120**) die Serveranfragedaten an den Geheimdaten speichernden Server (**130**) übertragen, ohne die schwach geheimen Daten zu gefährden. Der Geheimdaten speichernde Server (**130**) berechnet für mindestens einen der Geheimdaten speichernden Server (**130**) Serverantwortdaten. Die Serverantwortdaten enthüllen ohne Kenntnis der schwach geheimen Daten und der kurzlebigen Benutzergeheimdaten keine Kenntnis zu den Servergeheimdaten. Im Ergebnis kann der Geheimdaten speichernde Server (**130**) die Serverantwortdaten an den Erzeugungs-Client (**120**) übertragen, ohne seine Servergeheimdaten zu gefährden. Der Erzeugungs-Client (**120**) berechnet eine Geheimkomponente für den Geheimdaten speichernden Server als Funktion der von diesem empfangenen Serverantwortdaten und der kurzlebigen Benutzergeheimdaten. Die Geheimkomponente ist eine Funktion der schwach geheimen Daten und der Servergeheimdaten, jedoch ist sie von den kurzlebigen Benutzergeheimdaten unabhängig. Dann berechnet der Erzeugungs-Client (**120**) die stark geheimen Daten des Benutzers als Funktion der Geheimkomponenten.

[0015] Bei einer weiteren Verfeinerung bestehen die schwach geheimen Daten aus einem Passwort PWD, und die Servergeheimdaten sind ganzzahlige Zufallszahlen $B(i)$, wobei i ein Index für die Server (**130**) ist. Der Erzeugungs-Client (**120**) berechnet (**534**) Serverantwortdaten, die den Wert $M = w^3$ beinhalten. Hierbei gilt $w = f(\text{schwach geheime Daten})$, wobei f eine Funktion ist, die ein Element einer Gruppe G erzeugt, und die kurzlebigen Benutzergeheimdaten enthalten eine ganzzahlige Zufallszahl a , für die eine entsprechende ganze Zahl a' in solcher Weise existiert, dass für alle x in der Gruppe G $x^{aa'} = x$ gilt. Die Gruppe G ist eine endliche Gruppe, in der Potenzierung effizient ist, jedoch das diskrete Logarithmusproblem rechenmäßig nicht ausführbar ist, beispielsweise die multiplikative Gruppe der Reihe der ganzen Zahlen modulo a zur Primzahl p , oder eine Gruppe von Punkten auf einer elliptischen Kurve über ein endliches Feld. Alle Potenzierungen werden in der Gruppe G berechnet. Der Geheimdaten speichernde Server (**130**) berechnet (**534**) Serverantwortdaten, die den Wert $c(i) = M^{B(i)}$ beinhalten. Dann berechnet (**534**) der Erzeugungs-Client (**120**) Geheimkomponenten entsprechend $K(i) = h(c(i)^a)$ wobei h eine Funktion ist. Dann werden die stark geheimen Daten des Benutzers als Funktion aller Geheimkomponen-

ten $K(i)$, beispielsweise als Exklusiv-ODER-Verknüpfung dieser Komponenten, berechnet (**536**).

[0016] Gemäß einer anderen Erscheinungsform der Erfindung werden die stark geheimen Daten wie folgt aus den schwach geheimen Daten wiedergewonnen (**400**, **600**). Ein Wiedergewinnungsclient (**220**) berechnet dann (**440**, **640**) die stark geheimen Daten des Benutzers, die eine Funktion der schwach geheimen Daten des Benutzers und von Servergeheimdaten sind, für mindestens zwei Geheimdaten speichernde Server (**130**). Bei einer bevorzugten Ausführungsform basiert die Berechnung auf Geheimkomponenten, wie oben beschrieben. Der Wiedergewinnungsclient (**220**) bestimmt auch (**450**, **650**) Beweisdaten, um zu beweisen (**460**, **660**), dass die stark geheimen Daten erfolgreich berechnet wurden (**401**, **601**), und er überträgt (**455**, **655**) die Beweisdaten an Verifikationsserver (**130**), die auch Geheimdaten speichernde Server sein können, es aber nicht müssen. Durch eine Gültigkeitsprüfung der Beweisdaten unter Verwendung der entsprechenden Verifikationsdaten können die Verifikationsserver (**130**) bestimmen (**460**, **660**), ob die stark geheimen Daten erfolgreich wiedergewonnen wurden (**401**, **601**), und sie können geeignete Maßnahmen ergreifen (**680**). Wenn es beispielsweise wahrscheinlich erscheint, dass eine Einheit (**220**), die keinen Zugriff auf die schwach geheimen Daten hat, versucht, die stark geheimen Daten wiederzugewinnen (**401**, **601**), kann der Verifikationsserver die Geheimdaten speichernden Server (**130**) dazu anweisen, ihre Teilnahme an jedem weiteren Wiedergewinnungsversuch zu stoppen (**400**, **600**). Alternativ kann der Verifikationsserver dafür zuständig sein, einen Teil einer digitalen Signatur zugunsten des Benutzers (**110**) zu erzeugen. In diesem Fall kann mit den Beweisdaten ein Nachrichtenüberblick einer zu signierenden Nachricht einhergehen, und der Verifikationsserver (**130**) erzeugt seinen Teil der digitalen Signatur nur dann, wenn eine gleichzeitige Präsentation mit angemessenen Beweisdaten erfolgt. Wenn zusätzliche Privatdaten unter Verwendung der stark geheimen Daten als Kryptografieschlüssel verschlüsselt wurden (**370**), kann der Wiedergewinnungsclient (**220**) zusätzlich die Privatdaten entschlüsseln (**470**).

[0017] Bei einer bevorzugten Ausführungsform werden die stark geheimen Daten wie folgt berechnet (**440**, **640**). Der Wiedergewinnungsclient (**220**) berechnet (**420**, **620**) Serveranfragedaten für mindestens einen, vorzugsweise mehr als einen, Geheimdaten speichernden Server. Die Serveranfragedaten sind eine Funktion der schwach geheimen Daten und kurzlebiger Benutzergeheimdaten, jedoch enthüllen sie ohne Kenntnis der kurzlebigen Benutzergeheimdaten keine wesentliche Information zu den schwach geheimen Daten. Die Serveranfragedaten werden an den Geheimdaten speichernden Server übertragen (**425**, **625**), der auf Grundlage derselben und seiner

Servergeheimdaten Serverantwortdaten berechnet (430, 630). Die Serverantwortdaten enthüllen ohne Kenntnis der Serveranfragedaten keine wesentliche Information zu den Servergeheimdaten. Die Serverantwortdaten werden an den Wiedergewinnungsclient (220) übertragen (435, 635), der die stark geheimen Daten des Benutzers unter Verwendung der Serverantwortdaten wiedergewinnt (440, 640).

[0018] Bei einer bevorzugten Ausführungsform, die der oben beschriebenen entspricht, erzeugt (624) der Wiedergewinnungsclient (220) zufallsmäßig kurzlebige Servergeheimdaten a , wobei es sich um eine ganze Zahl handelt, für die eine entsprechende ganze Zahl a' in solcher Weise existiert, dass für alle x in der Gruppe G $x^{aa'} = x$ gilt. Er berechnet (626) den Wert $M = w^a$ enthaltende Serveranfragedaten, wobei $w = f(\text{PWD})$ gilt, wie oben, und er überträgt (625) diese Daten an mindestens einen Geheimdaten speichernden Server (130). Jeder Geheimdaten speichernde Server (130) berechnet (630) Serverantwortdaten, die den Wert $c(i) = M^{b(i)}$ enthalten, und er überträgt (635) denselben an den Wiedergewinnungsclient (220) zurück. Der Wiedergewinnungsclient (220) berechnet (644) die Geheimkomponenten $K(i) = h(c(i)^a)$, wobei h dieselbe Funktion wie oben ist. Die Geheimkomponenten werden dann kombiniert (646), wie im Erzeugungs-Client (120), um die stark geheimen Daten wiederzugewinnen.

[0019] Diese Verfahren (300, 400, 500, 600) sind von Vorteil, da sie es einem Benutzer (110) erlauben, stark geheime Daten an vielen Wiedergewinnungsclients (220) aus schwach geheimen Daten wiederzugewinnen (400, 600). Diese Verfahren sind im Wesentlichen resistent gegen Angriffe, einschließlich Angriffen auf die Server (130) oder gewollten Beeinträchtigungen derselben. Ferner werden gemäß der Erfindung die obigen Verfahren durch Software und/oder Hardware (100, 200) implementiert.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0020] Diese und andere detaillierteren und spezielleren Aufgaben und Merkmale der Erfindung werden in der folgenden Beschreibung vollständig offenbart, wobei auf die beigefügten Zeichnungen Bezug genommen wird.

[0021] Fig. 1 ist ein Blockdiagramm eines Initialisierungssystems (100) gemäß der Erfindung;

[0022] Fig. 2 ist ein Blockdiagramm eines Wiedergewinnungssystems (200) gemäß der Erfindung;

[0023] Fig. 3 ist ein Ereignisverlauf zum Veranschaulichen eines beispielhaften Verfahrens (300) zum Initialisieren des Systems 100, gemäß der Erfindung;

[0024] Fig. 4 ist ein Ereignisverlauf zum Veranschaulichen eines beispielhaften Verfahrens (900) zum Wiedergewinnen stark geheimer Daten, das unter Verwendung des Verfahrens 300 initialisiert wurde;

[0025] Fig. 5 ist ein Ereignisverlauf zum Veranschaulichen eines bevorzugten Verfahrens (500) zum Initialisieren des Systems 100 unter Verwendung von Potenzierung in einer Gruppe G , gemäß der Erfindung;

[0026] Fig. 6 ist ein Ereignisverlauf zum Veranschaulichen eines bevorzugten Verfahrens (600) zum Wiedergewinnen stark geheimer Daten, das unter Verwendung des Verfahrens 500 initialisiert wurde.

DETAILLIERTE BESCHREIBUNG DER BEVORZUGTEN AUSFÜHRUNGSFORMEN

[0027] Die Fig. 1 und Fig. 2 sind Blockdiagramme zum Veranschaulichen von Systemen 100 und 200 gemäß der Erfindung. Aus Gründen, die unten ersichtlich werden, wird das System 100 als "Initialisierungssystem" bezeichnet, und das System 200 wird als "Wiedergewinnungssystem" bezeichnet.

[0028] Das Initialisierungssystem 100 verfügt über einen Benutzer 110, ein Benutzerterminal 120, eine Anzahl von Servern 130A-130N (Gemeinsamserver 130) und wahlweise auch über ein Speichermedium 190. Beim Benutzer 110 kann es sich um eine Einzelperson, eine Gruppe von Einzelpersonen, eine juristische Person wie eine Firma, einen Computer oder dergleichen handeln. Das Benutzerterminal 120, das als "Erzeugungs-Client" 120 bezeichnet wird, ist typischerweise irgendein Typ einer Vorrichtung auf Computerbasis. Zu Beispielen gehören PCs, Computerworkstations sowie digitale Funktelefone. Die Server 130 sind ebenfalls typischerweise Vorrichtungen auf Computerbasis. In dieser Beschreibung werden sie aufgrund der Rolle, die sie spielen, als "Server" bezeichnet, jedoch soll diese Terminologie nicht bedeuten, dass es sich notwendigerweise um Computer der Serverklasse handeln würde. Mindestens einer der Server 130, und möglicherweise alle, sind Geheimdaten speichernde Server. Die von den Geheimdaten speichernden Servern gespielte Rolle wird unten vollständiger beschrieben. Bei bestimmten Ausführungsformen kann ein einzelner Server 130 vorhanden sein. Bei alternativen Ausführungsformen sind zwei oder mehr Geheimdaten speichernden Server 130 bevorzugt. Bei Ausführungsformen, die mehrere Geheimdaten speichernde Server 130 nutzen, werden diese vorzugsweise durch verschiedene Parteien kontrolliert, so dass, aus unten erörterten Gründen, keine einzelne Partei Zugriff auf alle Geheimdaten speichernden Server 130 hat. Zu Beispielen des Speichermediums 190 gehören ein Netz-

werk-Plattenlaufwerk, ein Verzeichnisdienst oder ein Dateiserver. Der Benutzer **110**, die Server **130** und das Speichermedium **140** sind jeweils mit dem Erzeugungs-Client **120** verbunden. Diese Verbindungen können über eine beliebige Anzahl von Maßnahmen erfolgen, einschließlich über Computernetzwerke wie das Internet oder durch drahtlose Verbindungen. Die Verbindungen müssen nicht dauerhaft oder beständig sein. Tatsächlich führt, wie es unten beschrieben wird, der Erzeugungs-Client **120** eine spezielle Aufgabe durch, und wenn diese einmal abgeschlossen ist, besteht für die anderen Komponenten kein Bedarf, weiterhin mit dem Erzeugungs-Client **120** zu kommunizieren.

[0029] Das Wiedergewinnungssystem **200** ist dem Initialisierungssystem **100** mit der Ausnahme ähnlich, dass der Erzeugungs-Client **120** durch ein anderes Benutzerterminal **220** ersetzt ist, das als Wiedergewinnungsclient **220** bezeichnet wird. Der Wiedergewinnungsclient **220** kann dieselbe körperliche Vorrichtung wie der Erzeugungs-Client **120** sein, was aber nicht der Fall sein muss. Zu Beispielen von Wiedergewinnungsclients **220** gehören PCs, digitale Telefonzellen, digitale Funktelefone oder andere drahtlose Vorrichtungen, und Smartcards.

[0030] Die Systeme **100** und **200** implementieren das folgende Funktionsvermögen. Der Benutzer **110** verfügt über stark geheime Daten, die er vom Wiedergewinnungsclient **220** aus verwenden möchte, wobei "stark" Daten kennzeichnet, bei denen es nicht ausführbar ist, sie durch erschöpfendes Raten herzu-leiten, und "geheim" bedeutet, dass es sich um Daten handelt, für die niemand anderer als der Geheimnisinhaber (beispielsweise der Benutzer, wenn es sich um stark geheime Daten desselben handelt) in der Lage sein sollte, sie zu bestimmen. Jedoch ist der Wiedergewinnungsclient **220** ein Benutzerterminal, das a priori keinen Zugriff auf die stark geheimen Daten des Benutzers **110** haben kann oder hat. Ferner kennt der Benutzer **110** seine stark geheimen Daten nicht direkt, so dass er beispielsweise dieselben nicht einfach in den Wiedergewinnungsclient **220** eingeben kann. Demgemäß muss der Wiedergewinnungsclient **220** auf irgendeine Weise die stark geheimen Daten des Benutzers **110** regenerieren oder wiedergewinnen, und er muss dies auf sichere Weise ausführen, um die starke Geheimhaltung der Daten aufrecht zu erhalten. Der Benutzer **110** kennt bestimmte schwach geheime Daten, wobei "schwach" bedeutet, dass die Daten innerhalb einer mäßigen Anzahl von Versuchen korrekt erraten werden können. Beispielsweise kann es sich um ein vom Benutzer spezifiziertes Passwort handeln. Der Benutzer **110** gibt diese schwach geheimen Daten in den Wiedergewinnungsclient **220** ein, und dann führt das System **200**, auf Grundlage dieser schwach geheimen Daten des Benutzers **110**, eine sichere Wiedergewinnung der stark geheimen Daten aus, um es so dem Benutzer

110 zu ermöglichen seine stark geheimen Daten vom Wiedergewinnungsclient **220** aus zu verwenden. Das System **100** und der Erzeugungs-Client **120** führen auf Grundlage der stark geheimen Daten und der schwach geheimen Daten verschiedene Initialisierungsschritte aus, so dass das System **200** und der Wiedergewinnungsclient **220** später die stark geheimen Daten aus den schwach geheimen Daten des Benutzers **110** sicher wiedergewinnen können. Die Server **130** unterstützen diese Prozesse.

[0031] Bei einer bevorzugten Ausführungsform handelt es sich bei den stark geheimen Daten um einen Kryptografieschlüssel, der in einem symmetrischen Kryptografiesystem dazu verwendet wird, Privatdaten des Benutzers **110**, die einer Privatschlüssel enthalten können, zu verschlüsseln, und bei den schwach geheimen Daten handelt es sich um ein Passwort. Die verschlüsselten Privatdaten werden im Speichermedium **140** gespeichert. Wenn es der Benutzer **110** wünscht, seine Privatdaten über den Wiedergewinnungsclient **220** zu nutzen, liefert er sein Passwort an diesen. Dann führt der Wiedergewinnungsclient **220** eine Wiedergewinnung des Kryptografieschlüssels aus, der dazu verwendet wird, die verschlüsselten Privatdaten des Benutzers zu entschlüsseln. Der Benutzer **110** kann dann seine Privatdaten nach Wunsch verwenden, beispielsweise unter Verwendung seines Privatschlüssels, um Nachrichten im Wiedergewinnungsclient **220** digital zu signieren.

[0032] Die [Fig. 3](#) und [Fig. 4](#) zeigen Ereignisverläufe zum Veranschaulichen beispielhafter Verfahren **300** und **400** zum Ausführen einer Initialisierung des Systems **100** bzw. zum Wiedergewinnen der stark geheimen Daten aus schwach geheimen Daten des Benutzers **110**. Das Verfahren **300** verwendet das System **100**, und das Verfahren **400** verwendet das System **200**. Jedes der gestrichelten Kästchen **110**, **120**, **130**, **140** und **220** repräsentiert eine der Komponenten des Systems **100** oder des Systems **200**. Die Kästchen mit durchgezogenen Strichen repräsentieren verschiedene Schritte bei den zwei Verfahren **300** und **400**. Der Ort eines Kästchens mit durchgezogenen Strichen innerhalb eines gestrichelten Kästchens zeigt allgemein an, dass der spezielle Schritt durch diese spezielle Komponente ausgeführt wird. Beispielsweise befindet sich in der [Fig. 3](#) der Schritt **310** im gestrichelten Kästchen für den Erzeugungs-Client **120**. Dies zeigt allgemein an, dass der Erzeugungs-Client **120** den Schritt **310** ausführt. Die Verfahren werden vorzugsweise durch Software implementiert, die auf den verschiedenen Komponenten in jedem System läuft, möglicherweise unterstützt durch Hardwaremodule, jedoch können die in den [Fig. 3](#) und [Fig. 4](#) dargestellten Verarbeitungsmodule auch durch Hardware und/oder Firmware implementiert werden.

[0033] Es wird zunächst auf die [Fig. 3](#) Bezug genommen, gemäß der das Initialisierungsverfahren **300** in drei allgemeine Stadien **301-303** unterteilt werden kann.

[0034] Im Erzeugungsstadium **301** bestimmt das System **100** die stark geheimen Daten des Benutzers, die bei dieser Ausführungsform als K bezeichnet werden. Dieses Stadium **301** beinhaltet Schritte, die es dem System **200** ermöglichen, später die stark geheimen Daten K wiederzugewinnen. Im Verifiziereinstellstadium **302** führt das System **100** Schritte aus, die es ihm ermöglichen, die anschließende erfolgreiche Rückgewinnung der stark geheimen Daten K zu verifizieren. Im Speicherstadium **303** verwendet das System **100** die stark geheimen Daten K zum Verschlüsseln von Privatschlüsseln für den Benutzer (beispielsweise den Privatschlüssel des Benutzers oder ein digitales Zertifikat), zur späteren Rückgewinnung durch den Wiedergewinnungsclient **420**. Nicht alle Implementierungen nutzen die Stadien **301-303**, jedoch sind sie bei diesem Beispiel enthalten, um verschiedene Gesichtspunkte der Erfindung zu veranschaulichen.

[0035] Im Erzeugungsstadium **303** beginnt der Erzeugungs-Client **120** damit, den Benutzer **110** als berechtigten Benutzer für das Benutzerkonto U zu identifizieren **310**. Dazu können Kommunikationsvorgänge mit anderen Systemen, wie einem Authentifizierungsserver, gehören.

[0036] Der Benutzer **110** und/oder der Erzeugungs-Client **120** bestimmt **320** die schwach geheimen Daten des Benutzers **110**, die hier als PWD bezeichnet werden. Bei den schwach geheimen Daten handelt es sich typischerweise bei dieser Ausführungsform um ein Benutzerpasswort, und diese Beschreibung charakterisiert sie als solche. Jedoch können die stark geheimen Daten andere Formen einnehmen, und/oder sie können ganz oder teilweise von einer anderen Quelle herrühren, wie biometrische Signaturdaten und/oder Daten, die auf einer körperlichen Karte oder einer dem Benutzer **110** zugeordneten Einrichtung gespeichert sind. In der [Fig. 3](#) ist es dargestellt, dass die Erzeugung **320** der schwach geheimen Daten sowohl den Benutzer **110** als auch den Erzeugungs-Client **120** überdeckt. Dies, da jeder derselben in verschiedenen Ausmaßen, abhängig von der speziellen Implementierung, teilhaben kann. Beispielsweise schlägt der Benutzer **110**, bei einer Vorgehensweise, ein Passwort PWD vor, und der Erzeugungs-Client **120** akzeptiert dieses, oder er weist es zurück, was davon abhängt, ob es bestimmte Antiratekriterien erfüllt. Bei einer anderen Vorgehensweise erzeugt der Erzeugungs-Client **120** das Passwort PWD (beispielsweise unter Verwendung eines Zufallsprozesses), und dann weist er es dem Benutzer **110** zu. In jedem Fall haben, am Ende des Schrittes **320**, sowohl der Benutzer **110** als auch

der Erzeugungs-Client **120** Zugriff auf das Passwort PWD.

[0037] Servergeheimdaten $b(i)$ für den Benutzer **110** werden für einige, und möglicherweise alle, der Server $S(i)$ erstellt **340**, wobei i ein Index für die Server ist. Die Server **130**, für die Servergeheimdaten erstellt sind, werden als Geheimdaten speichernde Server bezeichnet. Analog zum Schritt **320** ist dargestellt, dass der Schritt **340** den Erzeugungs-Client **120** und die Geheimdaten speichernden Server $S(i)$ abdeckt, da jeder derselben in verschiedenem Ausmaß, abhängig von der speziellen Implementierung, Teil haben kann, wie dies unten weiter beschrieben wird. Wenn eine spezielle Implementierung verlangt, dass der Erzeugungs-Client **120** und Geheimdaten speichernde Server $S(i)$ Nachrichten austauschen, um die Servergeheimdaten zu erstellen **340**, ist es wesentlich, dass diese Nachrichten in ihrer Unversehrtheit geschützt werden und die Quelle jeder Nachricht authentifiziert wird, um die Sicherheit des Gesamtprotokolls aufrechtzuerhalten. Beispielsweise können der Erzeugungs-Client und Geheimdaten speichernde Server $S(i)$ Nachrichten über einen sicheren Kanal austauschen. Am Ende des Schritts **340** hat jeder Geheimdaten speichernde Server $S(i)$ Zugriff auf seine entsprechenden Servergeheimdaten $b(i)$, und typischerweise speichert **345** er sie auch für zukünftigen Gebrauch bei der Wiedergewinnung der stark geheimen Daten des Benutzers auf sichere Weise. Der Erzeugungs-Client **120** kann, abhängig von der speziellen Implementierung, ebenfalls Zugriff auf die Servergeheimdaten $b(i)$ haben. Jedoch würde er in diesem Fall typischerweise die Servergeheimdaten $b(i)$ nur als Teil der Initialisierung bei **100** verwenden und sie dann löschen. Der Erzeugungs-Client **120** bewahrt die Servergeheimdaten $b(i)$ nicht für zukünftigen Gebrauch auf.

[0038] Der Erzeugungs-Client **120** berechnet **330** die stark geheimen Daten K des Benutzers. Die stark geheimen Daten K sind eine Funktion der schwach geheimen Daten PWD des Benutzers und der Servergeheimdaten $b(i)$. Erneut deckt der Schritt **330** sowohl den Erzeugungs-Client **120** als auch die Server $S(i)$ ab, da, abhängig von der Implementierung, jeder zur Berechnung beitragen kann, die dazu erforderlich ist, die schwach geheimen Daten PWD des Benutzers und die Servergeheimdaten $b(i)$ in die stark geheimen Daten K zu wandeln. Jedoch hat, obwohl die Geheimdaten speichernden Server $S(i)$ Zwischenwerte berechnen können, nur der Erzeugungs-Client **120** Zugriff auf die endgültigen stark geheimen Daten K.

[0039] Im Verifiziereinstellstadium **302** werden Verifikationsdaten $v(i)$ für einige und möglicherweise alle der Server $S(i)$ bestimmt **350**, und vorzugsweise werden sie auch durch jeden Server $S(i)$ abgespeichert **355**. Die Server, für die Verifikationsdaten erstellt

werden, werden als Verifikationsserver **130** bezeichnet. Die Verifikationsdaten $v(i)$ ermöglichen es den Verifikationsservern $S(i)$, zu verifizieren, ob ein Wiedergewinnungsclient **220** die stark geheimen Daten des Benutzers erfolgreich wiedergewonnen hat. Bei einer bevorzugten Ausführungsform sind die Geheimdaten speichernden Server auch die Verifikationsserver, obwohl dies nicht notwendigerweise der Fall ist. Alternativ können die Verifikationsserver von den Geheimdaten speichernden Servern körperlich getrennt sein, jedoch kann ein Verifikationsserver vorhanden sein, der jedem Geheimdaten speichernden Server entspricht. Zu Redundanz Zwecken ist es bevorzugt, über mindestens zwei Verifikationsserver zu verfügen. Die Verifikationsdaten werden so ausgewählt, dass es dem Verifikationsserver rechenermäßig unmöglich ist, die schwach geheimen Daten alleine durch Zugriff auf seine Verifikationsdaten zu bestimmen. Bei einer Vorgehensweise bestimmt **350** der Erzeugungs-Client **120** die Verifikationsdaten, die dann an den Server **130** übertragen werden. Bei einer alternativen Vorgehensweise bestimmt jeder Server **130** seine eigenen Verifikationsdaten. Analog zum Schritt **340** ist es, wenn es eine spezielle Implementierung verlangt, dass der Erzeugungs-Client **120** und der Server **130** Nachrichten austauscht, um die Verifikationsdaten zu bestimmen **350**, wesentlich, dass diese Nachrichten hinsichtlich der Unversehrtheit geschützt werden und die Quelle jeder Meldung authentifiziert wird.

[0040] Im Speicherstadium **303** verschlüsselt sie **370** der Erzeugungs-Client **120** zusätzlich anderen Daten für den Benutzer, die als Privatdaten des Benutzers bezeichnet werden. Bei einer bevorzugten Ausführungsform werden die stark geheimen Daten K als Kryptografieschlüssel in einem symmetrischen kurzlebigen System verwendet. Beispielsweise könnten die Privatdaten der Privatschlüssel des Benutzers, Geheimdaten, die der Benutzer und ein Anwendungsserver gemeinsam haben, die Kontonummern von Kreditkarten des Benutzers oder andere private oder geheime Daten sein, die der Benutzer vom Wiedergewinnungsclient **220** aus zu verwenden wünscht. Die verschlüsselten Privatdaten EPD werden zur späteren Wiedergewinnung im Speichermedium **140** gespeichert **375**. Auf das Speichermedium **140** kann typischerweise in weitem Umfang zugegriffen werden; Die Privatdaten des Benutzers sind geschützt, da sie in verschlüsselter Form abgespeichert sind. Bei alternativen Ausführungsformen müssen die stark geheimen Daten K auf andere Arten dazu verwendet werden, die Privatdaten des Benutzers sicher zu speichern.

[0041] Es wird nun auf die [Fig. 4](#) Bezug genommen, gemäß der das Wiedergewinnungsverfahren **400** ebenfalls in drei allgemeine Stadien **401-403** unterteilt werden kann, die den Stadien **301-303** des Verfahrens **300** entsprechen, wobei nicht alle bei allen

Implementierungen erforderlich sind. Im Stadium **401** führt der Wiedergewinnungsclient **220** eine Wiedergewinnung der stark geheimen Daten K des Benutzers unter Unterstützung durch die Geheimdaten speichernden Server **130** aus. Im Stadium **402** bestimmen ein oder mehrere Verifikationsserver **130**, ob der Wiedergewinnungsclient **220** die stark geheimen Daten K erfolgreich wiedergewonnen hat. Im Stadium **403** führt der Wiedergewinnungsclient **220** auch eine Wiedergewinnung der im Speichermedium **140** gespeicherten Privatdaten des Benutzers aus. Wiederum wird das beispielhafte Verfahren **400** ausgewählt, um verschiedene Gesichtspunkte der Erfindung zu veranschaulichen, die nicht alle zum Ausführen derselben erforderlich sind.

[0042] Der Wiedergewinnungsclient **220** führt eine Wiedergewinnung **401** der stark geheimen Daten des Benutzers **110** wie folgt aus. Der Benutzer **140** gibt seine Benutzerkontokennung U und schwach geheime Daten PWD in den Wiedergewinnungsclient **220** ein **410**. Der Wiedergewinnungsclient **220** berechnet **420** Serveranfragedaten für jeden der Server $S(i)$, und er überträgt **425** dieselben an die Geheimdaten speichernden Server. Die Serveranfragedaten sind eine Funktion der schwach geheimen Daten PWD und kurzlebiger Benutzergeheimdaten a , so dass die Ausgangsinformation der Funktion keine Information zu den schwach geheimen Daten an eine Partei Preis gibt, die die kurzlebigen Benutzergeheimdaten a nicht kennt. Auf die empfangenen Serveranfragedaten hin berechnet **430** jeder der Geheimdaten speichernden Server $S(i)$ Serverantwortdaten, bei denen es sich um eine Funktion der Serveranfragedaten und der Servergeheimdaten $b(i)$ handelt, und er überträgt **435** die Serverantwortdaten an den Wiedergewinnungsclient **220** zurück. Dann berechnet **440** der Wiedergewinnungsclient **220** die stark geheimen Daten K des Benutzers als Funktion der empfangenen Serverantwortdaten. Wie bereits beschrieben, sind die stark geheimen Daten eine Funktion der schwach geheimen Daten des Benutzers und der Servergeheimdaten. Der Wiedergewinnungsclient **220** hat direkten Zugriff auf die schwach geheimen Daten des Benutzers, jedoch keinen direkten Zugriff auf die Servergeheimdaten. Jedoch enthalten die Serverantwortdaten eine Abhängigkeit von den Servergeheimdaten; so hat der Wiedergewinnungsclient **220** indirekten Zugriff auf die Servergeheimdaten, und er kann die stark geheimen Daten des Benutzers wiedergewinnen, ohne dass er direkten Zugriff auf die Servergeheimdaten benötigt.

[0043] Im Wiedergewinnungsstadium **403** ruft der Wiedergewinnungsclient **220** die verschlüsselten Privatdaten EPD des Benutzers ab **475**, und er entschlüsselt **470** dieselben unter Verwendung der wiedergewonnenen stark geheimen Daten K . Auf diese Weise führt der Wiedergewinnungsclient **220** auch eine Wiedergewinnung der Privatdaten des Benut-

zers, wie des Privatschlüssels des Benutzers, aus.

[0044] Mit dem Ende der Periode des berechtigten Gebrauchs der stark geheimen Daten K und beliebiger anderer wiedergewonnener Privatdaten (beispielsweise am Ende einer Onlinesitzung des Benutzers unter Verwendung des Wiedergewinnungsclients **220**) werden die Kopie von K und der anderen wiedergewonnenen Privatdaten im Wiedergewinnungsclient **220** vorzugsweise zerstört.

[0045] Im Verifizierstadium **402** bestimmt **450** der Wiedergewinnungsclient **220** Beweisdaten $d(i)$, um mindestens einem Verifikationsserver (vorzugsweise mindestens zwei Verifikationsservern) zu beweisen, dass die stark geheimen Daten durch den Wiedergewinnungsclient **220** erfolgreich wiedergewonnen wurden. Die Beweisdaten $d(i)$ werden an jeden der Verifikationsserver übertragen **455**. Jeder Verifikationsserver kann dann verifizieren **460**, ob diese spezielle Instanz des Wiedergewinnungsprozesses **400** die stark geheimen Daten K des Benutzers erfolgreich wiedergewonnen hat, und er kann geeignete Maßnahmen ergreifen. Beispielsweise kann, bei einer Ausführungsform, ein Verifikationsserver **130** für einen Teil des Prozesses zum Erzeugen einer digitalen Signatur zugunsten des Benutzers **110** zuständig sein. In diesem Fall können die Beweisdaten von einem Nachrichtenüberblick der digital zu signierenden, vom Benutzer herrührenden Nachricht begleitet werden. Auf das Verifizieren der Beweisdaten hin erzeugt der Verifikationsserver **130** seine Komponente der digitalen Signatur und überträgt diese an den Client zurück. Der Client bestimmt die vollständige digitale Signatur auf Grundlage der Komponenten, die er empfängt.

[0046] Gemäß einem anderen Beispiel kann der Verifikationsserver auch ein Geheimdaten speichernder Server sein. Dieser Server kann, auf Grundlage des Verlaufs nicht erfolgreicher Versuche in der Vergangenheit, bestimmen, dass eine Partei, die die schwach geheimen Daten nicht kennt, versucht, die stark geheimen Daten zu regenerieren. Demgemäß kann der Geheimdaten speichernde Server die Teilnahme an weiteren Wiedergewinnungsversuchen zurückweisen oder andere Maßnahmen ergreifen. Bei einer Vorgehensweise bewahren die Geheimdaten speichernden Server einen Verlauf aller Versuche zum Regenerieren der stark geheimen Daten K für jedes Benutzerkonto auf, und wenn für irgendein Konto übermäßig viele fehlgeschlagene Versuche vorliegen, drosseln sie weitere Regenerierungsversuche für dieses Konto, und/oder schließen sie aus, bis entweder das Passwort geändert ist, oder ein örtlicher Administrator bestimmt, dass die fehlgeschlagenen Versuche keine Gefahr für das Konto darstellen.

[0047] Es können verschiedene Verifikationstypen verwendet werden. Bei einer Vorgehensweise unter

Verwendung statischer Verifikationsdaten bilden die Verifikationsdaten $v(i)$ eine unidirektionale Funktion einer oder mehrerer Einzeldaten, die die stark geheimen Daten K enthalten. Die Verifikationsdaten werden vom Erzeugungs-Client **120** oder irgendeiner anderen vertrauenswürdigen Partei berechnet und ein Teil des Initialisierungsprozesses **300** an jeden Verifikationsserver gesendet und in diesem gespeichert. Vorzugsweise werden für jeden Verifikationsserver **130** verschiedene Verifikationsdaten berechnet, beispielsweise durch Anwenden einer Hashfunktion auf einen Datenstring aus den stark geheimen Daten K in Verkettung mit einer eindeutigen, jedoch nicht geheimen Kennung für den speziellen Server **130**. Im Verifizierstadium **402** berechnet **450** der Wiedergewinnungsclient **220** die Beweisdaten durch Neuberechnen derselben unidirektionalen Funktion der regenerierten stark geheimen Daten. Wenn die durch den Wiedergewinnungsclient **220** berechneten Beweisdaten mit den durch den Server gespeicherten Verifikationsdaten übereinstimmen, ist die Wiedergewinnung der stark geheimen Daten verifiziert.

[0048] Bei einer Variante dieser Vorgehensweise berechnet der Wiedergewinnungsclient **220** erneut Verifikationsdaten als unidirektionale Funktion der stark geheimen Daten K, und dann berechnet er die Beweisdaten als unidirektionale Form eines oder mehrerer Einzeldatenwerte, die die neu berechneten Verifikationsdaten und eine nicht geheime Ad-hoc-Information, wie eine Zeitmarkierung oder eine Ad-hoc-Information enthalten, die zuvor entweder gemeinsam mit den Serverantwortdaten (beispielsweise bei einem Geheimdaten speichernden Server) oder mit einer gesonderten Nachricht vom Server an den Wiedergewinnungsclient gesendet wurde. Der Verifikationsserver **130** berechnet seine eigenen Beweisdaten durch Anwenden derselben unidirektionalen Funktion auf seine Kopie der Verifikationsdaten und der Ad-hoc-Information. Wenn die durch den Wiedergewinnungsclient **220** berechneten Beweisdaten mit den durch den Verifikationsserver **130** berechneten Beweisdaten übereinstimmen, ist die Wiedergewinnung der stark geheimen Daten verifiziert. Die Ad-hoc-Information erlaubt es dem Server, zu bestätigen, dass die Beweisdaten frisch sind und nicht frühere Beweisdaten für denselben Benutzer wiederverwendet werden. Anders gesagt, unterscheidet die Ad-hoc-Information die aktuellen Beweisdaten von anderen Instanzen derselben für denselben Benutzer.

[0049] Bei einer anderen Verifizierungsvorgehensweise wird davon ausgegangen, dass die im Speichermedium **140** gespeicherten Privatdaten des Benutzers Privatdaten (beispielsweise einen Benutzer) eines asymmetrischen Beweissystems enthalten, für das entsprechende öffentliche Daten existieren. Das asymmetrische Beweissystem zeigt die Eigenschaft, dass eine Partei, die Zugriff auf die Privatdaten hat,

einer zweiten Partei mit Zugriff auf die öffentlichen Daten beweisen kann, dass sie Zugriff auf die Privatdaten hat, ohne dass sie der zweiten Partei die Privatdaten offenbaren müsste. Das asymmetrische Beweissystem könnte ein digitales Signatursystem, wie RSA oder DSA, ein kenntnisfreies Beweissystem, bei dem der Besitz von Information verifiziert werden kann, ohne dass irgendein Teil dieser Information preisgegeben wird, oder ein anderer Typ eines asymmetrischen Beweissystems sein. Um die Erfindung zu beschreiben, wird ein digitales Signatursystem angenommen. In diesem Fall kann der Wiedergewinnungsclient **220** Beweisdaten erzeugen, die von den verschlüsselten Privatdaten abhängen, um so die erfolgreiche Wiedergewinnung der Privatdaten zu beweisen, und um damit auch die erfolgreiche Wiedergewinnung der stark geheimen Daten zu beweisen (da die stark geheimen Daten dazu erforderlich sind, die Privatdaten vollständig zu entschlüsseln).

[0050] Wenn beispielsweise die Privatdaten aus dem privaten Digitalsignaturschlüssel des Benutzers bestehen, könnten die Beweisdaten eine Nachricht enthalten, die vorzugsweise Ad-hoc-Information enthält, um die Erkennung von Wiederverwendungen zu ermöglichen, die unter Verwendung des Privatschlüssels digital signiert wurde. Wenn die Verifikationsdaten aus dem entsprechenden öffentlichen Schlüssel besteht, könnten sie dazu verwendet werden, die erfolgreiche Wiedergewinnung des Privatschlüssels zu verifizieren. Zum Verifizieren **460** der erfolgreichen Wiedergewinnung der stark geheimen Daten würde es dann gehören, zu verifizieren, dass der Privatschlüssel zum digitalen Signieren der Nachricht verwendet wurde. Wenn Ad-hoc-Information verwendet wird, würde es zum Verifizieren der Frische der Privatdaten gehören, zu verifizieren, dass die digital signierte Nachricht die Ad-hoc-Information enthält. Als Alternative können die Beweisdaten eine Funktion anderer Benutzerdaten sein, die der Verifikationsserver authentifizieren kann oder als vom Benutzer herrührend erkennen kann. Andere Variationen des Verifizierschritts können auf anderen Vorgehensweisen beruhen, beispielsweise auf kenntnisfreien Beweisen (siehe beispielsweise J. Nechvatal in "Public Key Cryptography", G.J. Simmons (Ed.), Contemporary Cryptology: The Science of Information Integrity (New York: IEEE Press, 1992), S. 107-126).

[0051] Es sind andere Variationen der Verfahren **300** und **400** ersichtlich. Um jedoch die Unterstützung gegen Angriffe aufrechtzuerhalten, einschließlich eines Eingriffs in die Server, sollte jedes erfolgreiche Protokoll vorzugsweise die folgenden Attribute enthalten. Als Erstes kann eine Beobachtung beliebiger oder aller Nachrichten durch einen Abhörer demselben nicht ausreichend Information liefern, um entweder die schwach geheimen Daten PWD oder die stark geheimen Daten K in machbarer Weise herzuleiten.

Zweitens erlaubt es die Kenntnis von irgendetwas, das weniger als die Gesamtheit der Servergeheimdaten von einer vorbestimmten Anzahl von Geheimdaten speichernden Servern entspricht, einer beliebigen Partei nicht, in machbarer Weise entweder die schwach geheimen Daten PWD oder die stark geheimen Daten K herzuleiten. "Eine beliebige Partei" beinhaltet die Server selbst. D.h., dass ein Server nicht in machbarer Weise entweder schwach geheime Daten PWD oder die stark geheimen Daten K herleiten kann, solange nicht eine vorbestimmte Anzahl von Geheimdaten speichernden Servern dadurch gemeinsame Sache machen, dass sie entweder ihre Servergeheimdaten offenbaren oder das Konto nicht drosseln oder sperren, wenn beim Ausführen des Protokolls übermäßig viele fehlgeschlagene Versuche auftreten. Im Ergebnis sind die Verfahren **300** und **400** von Vorteil, da sie gegen viele Angriffsarten resistent sind, einschließlich eines Eingriffs in einen Server, wenn die stark geheimen Daten von Servergeheimdaten mindestens zwei ihrer Geheimdaten speichernden Server abhängen.

[0052] Die [Fig. 5](#) und [Fig. 6](#) sind Ereignisverläufe zum Veranschaulichen bevorzugter Ausführungsformen **500** und **600** der Verfahren **300** bzw. **400**. Die bevorzugten Ausführungsformen beruhen auf Berechnungen innerhalb einer endlichen Gruppe G, in der die Potenzierung effizient ist, jedoch das diskrete Logarithmusproblem rechenmäßig nicht lösbar ist. Eine geeignete Gruppe G ist die multiplikative Gruppe der Reihe ganzer Zahlen modulo a zur Primzahl p, wobei p eine große Primzahl mit Eigenschaften ist, die sie als Diffie-Hellman-Modul geeignet macht. Insbesondere muss $p - 1$ minimal einen großen Primfaktor aufweisen. Eine bessere Einschränkung für p besteht darin, dass es sich um eine sichere Primzahl $p = 2q + 1$ handelt, wobei q eine Primzahl ist. Es sind auch andere zyklische Gruppen geeignet, einschließlich einer Gruppe von Punkten auf einer elliptischen Kurve über ein endliches Feld. Die Verfahren werden in einem Zusammenhang veranschaulicht, gemäß dem die stark geheimen Daten K anschließend als Kryptografieschlüssel verwendet werden, der dazu verwendet wird, den Privatschlüssel des Benutzers zu verschlüsseln, wobei jedoch, wie bereits bei den Verfahren **300** und **400** erörtert, die bevorzugten Verfahren **500** und **600** nicht auf diese spezielle Anwendung beschränkt sind. Bei diesem speziellen Beispiel funktioniert jeder Server **130** sowohl als Geheimdaten speichernder Server als auch als Verifikationsserver.

[0053] Es wird als Erstes auf die [Fig. 5](#) Bezug genommen, gemäß der das Initialisierungsverfahren **500** in die drei allgemeinen Stadien **501-503** zerlegt werden kann, die den Stadien **301-303** des Verfahrens **300** analog sind. Das Erzeugungsstadium **501** beginnt gleich wie das Erzeugungsstadium **301**. Der Erzeugungs-Client **120** authentifiziert **310** den Benut-

zer als legitimen Benutzer für das Konto U. Als Nächstes werden die schwach geheimen Daten PWD des Benutzers bestimmt **320**.

[0054] Servergeheimdaten $b(i)$ für den Benutzer **110** werden für jeden Geheimdaten speichernden Server $S(i)$ bestimmt **340**, wobei i ein Index für die Server **130** ist. Bei dieser Ausführungsform bestehen die Servergeheimdaten aus einer ganzzahligen Zufallszahl $b(i)$, vorzugsweise einer geraden Zahl, um vor Angriffen gegen kleine Untergruppen zu schützen, die bei Diffie-Hellman-Verfahren gut bekannt sind. Am Ende des Schritts **340** hat jeder Geheimdaten speichernde Server Zugriff auf seine Servergeheimdaten $b(i)$; der Erzeugungs-Client **120** kann Zugriff auf die Servergeheimdaten $b(i)$ haben, was jedoch nicht der Fall sein muss, und von der speziellen Implementierung abhängt. Bei einer Vorgehensweise, bei der sowohl ein Geheimdaten speichernder Server $S(i)$ als auch der Erzeugungs-Client **120** Zugriff auf die Servergeheimdaten $b(i)$ des Servers haben, werden die Servergeheimdaten $b(i)$ entweder durch den Erzeugungs-Client **120** oder den Server $S(i)$ erzeugt und der anderen Partei in verschlüsselter Form mitgeteilt. Alternativ werden die Servergeheimdaten dadurch berechnet, dass Zufallswerte von sowohl dem Erzeugungs-Client **120** als auch dem Geheimdaten speichernden Server $S(i)$ kombiniert werden, z.B. durch Diffie-Hellman-Schlüsselaustausch. Kommunikationsvorgänge werden vorzugsweise so geschützt, dass keine andere Partei $b(i)$ erfahren kann. Wenn andererseits der Erzeugungs-Client **120** keinen Zugriff auf die Servergeheimdaten $b(i)$ hat, kann jeder Geheimdaten speichernde Server $S(i)$ seine Servergeheimdaten $b(i)$ erzeugen, sie jedoch nicht gemeinsam mit dem Erzeugungs-Client **120** oder den anderen Servern $s(i)$ verwenden. Unabhängig vom Erzeugungsverfahren führt jeder Server typischerweise eine sichere Speicherung **345** seiner Servergeheimdaten für zukünftige Verwendung beim Regenerieren der stark geheimen Daten des Benutzers aus.

[0055] Der Erzeugungs-Client **120** berechnet **530** die stark geheimen Daten K wie folgt. Als Erstes berechnet **532** der Erzeugungs-Client **120** $w = f(\text{PWD})$, wobei f eine unidirektionale Funktion ist, die ein Element einer Gruppe G erzeugt. Als Nächstes wird für jeden Geheimdaten speichernden Server entsprechend der Beziehung $K(i) = h(w^{b(i)})$ eine Geheimkomponente $K(i)$ berechnet **534**. In diesem Ausdruck ist h eine unidirektionale Funktion, wie eine kurzlebige Hash-Funktion, die einen Wert mit Bits im Wesentlichen ohne Grundwert und ohne Korrelation erzeugt, wie er zur Verwendung als Schlüssel zur symmetrischen Verschlüsselung geeignet sein kann. Die Potenzierung $w^{b(i)}$ wird in der Gruppe G berechnet (d.h. bei diesem Beispiel ganze Zahlen modulo p). Abhängig von der Implementierung können die Geheimdaten speichernden Server an dieser Berechnung Teil

haben, was jedoch nicht der Fall sein muss. Es sei darauf hingewiesen, dass jede Geheimkomponente eine Funktion sowohl der schwach geheimen Daten des Benutzers als auch der stark geheimen Daten für den entsprechenden Geheimdaten speichernden Server ist.

[0056] Wenn beispielsweise der Erzeugungs-Client **120** direkte Kenntnis der Servergeheimdaten $b(i)$ hat, kann er die Geheimkomponente $K(i) = h(w^{b(i)})$ direkt berechnen **534**. Alternativ kann, wenn der Erzeugungs-Client **120** keinen Zugriff auf die Servergeheimdaten $b(i)$ hat, das folgende geschützte Transaktionsverfahren verwendet werden. Der Erzeugungs-Client **120** wählt kurzlebige Benutzergeheimdaten aus, bei denen es sich um eine ganzzahlige Zufallszahl a handelt, für die eine entsprechende ganze Zahl a' so existiert, dass für alle x in der Gruppe G $x^{aa'} = x$ gilt. Wenn beispielsweise G die Gruppe ganzer Zahlen modulo p ist, kann a so ausgewählt werden, dass es sich um ein Element der multiplikativen Gruppe der ganzen Zahlen modulo $p - 1$ handelt, und dann wäre a' der zugehörige multiplikative Kehrwert. Der Erzeugungs-Client berechnet und liefert eine Nachricht $M_i = w^a$ an jeden Server $S(i)$. Der Server $S(i)$ berechnet $c(i) = M_i^{b(i)} = w^{ab(i)}$, und er liefert $c(i)$ an den Erzeugungs-Client **120**. Der Erzeugungs-Client **120** berechnet den a entsprechenden Wert a' , und dann berechnet **534** er die Geheimkomponente $K(i) = h(c(i)^{a'}) = h(w^{b(i)})$. Diese Vorgehensweise gewährleistet, dass keine Geheimdaten einem Abhörer frei gelegt werden, wobei Verwendung der Verschlüsselungseigenschaften bei diskreter Potenzierung erfolgt.

[0057] Wenn einmal der Erzeugungs-Client **120** die Geheimkomponenten $K(i)$ berechnet hat, berechnet er **536** die stark geheimen Daten K als Funktion der Speicherkondensatoren $K(i)$ der teilhabenden Geheimdaten speichernden Server $S(i)$. Bei diesem Beispiel werden die stark geheimen Daten K entsprechend einer bitweisen Exklusiv-ODER-Verknüpfung berechnet **536**, also $K = K(1) \oplus K(2) \oplus \dots \oplus K(N)$, wobei $\oplus$ die Exklusiv-ODER-Verknüpfung bezeichnet. Binäre Addition und Verfahren zur gemeinsamen Nutzung von t -aus- N -Schwellenwertgeheimnissen sind zwei weitere Verfahren zum Kombinieren der Geheimkomponenten $K(i)$. Bei einem Verfahren zur gemeinsamen Nutzung von t -aus- N -Schwellenwertgeheimnissen existieren N Geheimdaten speichernde Server, jedoch können die stark geheimen Daten aus Wiedergewinnungsdaten von nur t derselben berechnet werden, wobei t kleiner als N ist. Es sind andere Verfahren ersichtlich.

[0058] Im Speicherstadium **503** werden die stark geheimen Daten K als Kryptografieschlüssel in einem symmetrischen Kryptografiesystem zum Verschlüsseln **370** von Privatdaten des Benutzers einschließlich eines Privatschlüssels Priv_U verwendet.

Die verschlüsselten Privatdaten, die als EPD bezeichnet werden und die den verschlüsselten Privatschlüssel enthalten, der als $E_K(\text{Priv}_U)$ bezeichnet wird, wobei E_K eine Verschlüsselung mit dem Schlüssel K bedeutet, wird im Speichermedium **140** gespeichert **375**.

[0059] Im Verifiziereinstellstadium **502** kann der dem Privatschlüssel, Priv_U , des Benutzers entsprechende öffentliche Schlüssel Pub_U , als Verifikationsdaten $v(i)$ verwendet werden. Bei dieser Ausführungsform spielt jeder Geheimdaten speichernde Server auch die Rolle eines Verifikationsservers $S(i)$, und er speichert **355** seine Verifikationsdaten $v(i) = \text{Pub}_U$, oder er hat zumindest Zugriff auf den öffentlichen Schlüssel. Bei einer alternativen Vorgehensweise sind die Verifikationsdaten $v(i) = h(K, \text{Id}(i))$, wobei $\text{Id}(i)$ eine eindeutige, aber öffentlich bekannte Kennung für den Server $S(i)$ ist und h eine unidirektionale Funktion wie eine Hash-Funktion ist. Der Einschluss von $\text{Id}(i)$ gewährleistet, dass für jeden Verifikationsserver verschiedene Verifikationsdaten erzeugt werden.

[0060] Es wird nun auf die [Fig. 6](#) Bezug genommen, gemäß der der Wiedergewinnungsprozess **600** ebenfalls drei Stadien **601-603** enthält. Im Stadium **601** führt der Wiedergewinnungsclient **220** eine Wiedergewinnung der stark geheimen Daten K des Benutzers mit Unterstützung der Geheimdaten speichernden Server auf Grundlage seiner schwach geheimen Daten PWD aus. Im Verifizierstadium **602** beweist der Wiedergewinnungsclient **220** den Verifikationsservern, dass er die stark geheimen Daten K erfolgreich wiedergewonnen hat. Im Stadium **603** führt der Wiedergewinnungsclient **220** eine Wiedergewinnung des Privatschlüssels, Priv_U , des Benutzers aus.

[0061] Die Wiedergewinnung **601** der stark geheimen Daten K beginnt damit, dass der Wiedergewinnungsclient **220** die Benutzerkontokennung U und das Passwort PWD vom Benutzer **110** empfängt **410**. Der Wiedergewinnungsclient **220** regeneriert dann die benötigten Speicherkondensatoren $K(i)$ unter Verwendung des oben beschriebenen geschützten Transaktionsverfahrens. Insbesondere berechnet **622** der Wiedergewinnungsclient **200** $w = f(\text{PWD})$, wobei f dieselbe unidirektionale Funktion ist, wie sie im Erzeugungsstadium **500** verwendet wird. Der Wiedergewinnungsclient **220** wählt **629** kurzlebige Benutzergeheimdaten aus, bei denen es sich um eine ganzzahlige Zufallszahl a handelt, für die eine entsprechende ganze Zahl a' in solcher Weise existiert, dass für alle x in der Gruppe G $x^{aa'} = x$ gilt. Wenn beispielsweise G die Gruppe der ganzen Zahlen Modulo p ist, kann a als Element der multiplikativen Gruppe der ganzen Zahlen modulo $p - 1$ ausgewählt werden, und a' wäre dann der zugehörige multiplikative Kehrwert. Dann berechnet **626** der Wiedergewinnungsclient die Serveranfragedaten $M_1 = w^a$ und überträgt

625 dieselben an den Server $S(i)$. Es sei darauf hingewiesen, dass die Serveranfragedaten M_1 eine Funktion sowohl der schwach geheimen Daten PWD als auch der kurzlebigen Benutzergeheimdaten a sind. Jedoch enthüllen die Serveranfragedaten M_1 ohne Kenntnis der kurzlebigen Benutzergeheimdaten a keine Information zu den schwach geheimen Daten PWD .

[0062] Der Server $S(i)$ empfängt die Serveranfragedaten M_1 . Der Server inkrementiert einen Zähler für nicht verifizierte Wiedergewinnungsversuche für das Benutzerkonto U und das aktuelle Passwort PWD , und er ermittelt **680**, ob es wahrscheinlich ist, dass eine Partei ohne Zugriff auf das Passwort versucht, die stark geheimen Daten wiederzugewinnen. Bei dieser Ausführungsform führt er dies dadurch aus, dass er ermittelt, ob die Anzahl der nicht verifizierten Wiedergewinnungsversuche einen Schwellenwert überschreitet. Wenn dies der Fall ist, sperrt der Server das Benutzerkonto U und verlässt den Wiedergewinnungsprozess **600**. Abhängig von den Eigenschaften der Gruppe G kann der Server auch verifizieren, dass die Serveranfragedaten M_1 erforderlichen strengen Eigenschaften genügen. Wenn die Serveranfragedaten M_1 nicht über die erforderlichen strengen Eigenschaften verfügen, gibt der Server den Wiedergewinnungsprozess auf. Wenn der Wiedergewinnungsprozess nicht aufgegeben wurde, berechnet **630** der Server die Serverantwortdaten $c(i) = M_1^{b(i)} = w^{ab(i)}$ und er liefert **635** $c(i)$ an den Wiedergewinnungsclient **220**. Der Server erzeugt **690** auch einen eindeutigen Index $n(i)$, oder Ad-hoc-Information, für diese Instanz des Wiedergewinnungsprozesses, und er überträgt die Ad-hoc-Information an den Wiedergewinnungsclient **220**. Der Server stellt eine Zustandsvariable ein, die die für die Ad-hoc-Information $n(i)$ anhängige Verifikation anzeigt. Bei einer bevorzugten Vorgehensweise überträgt der Server eine einzelne Nachricht, die sowohl auf den Serverantwortdaten $c(i)$ als auch der Ad-hoc-Information $n(i)$ beruht, an den Wiedergewinnungsclient **220**. Ähnlich den Serveranfragedaten M_1 sind die Serverantwortdaten $c(i)$ eine Funktion der Servergeheimdaten $b(i)$ für den Geheimdaten speichernden Server und die empfangenen Serveranfragedaten M_1 . Jedoch enthüllen die Serverantwortdaten $c(i)$ ohne Kenntnis der Serveranfragedaten M_1 , oder in gewisser Weise entsprechend, den schwach geheimen Daten PWD und den kurzlebigen Benutzergeheimdaten a keine Information zu den Servergeheimdaten $b(i)$.

[0063] Wenn der Wiedergewinnungsclient **220** die Nachricht vom Server empfangen hat, kann er den Wiedergewinnungsprozess **600** verlassen, wenn die empfangene Nachricht nicht über die erforderliche Stärke verfügt. Andernfalls berechnet **642** der Wiedergewinnungsclient **220** den a entsprechenden Wert a' . Dann berechnet er **644** die Geheimkomponente $K(i) = h(c(i)^a) = h(w^{b(i)})$. Es sei darauf hingewie-

sen, dass die Verwendung der kurzlebigen Benutzergeheimdaten a die Kommunikationsvorgänge zwischen dem Wiedergewinnungsclient **220** und dem Geheimdaten speichernden Server **130** resistent gegen Angriffe macht, die versuchen, die schwach geheimen Daten PWD oder Servergeheimdaten $b(i)$ herzuleiten. Jedoch ist die Geheimkomponente $K(i)$ eine Funktion sowohl der schwach geheimen Daten PWD als auch der Servergeheimdaten $b(i)$, jedoch ist sie unabhängig von den kurzlebigen Benutzergeheimdaten a . Schließlich berechnet **646** der Wiedergewinnungsclient **220** die stark geheimen Daten $K = K(1) \oplus K(2) \oplus \dots \oplus K(N)$. Der Wiedergewinnungsclient **220** kann dann den Privatschlüssel Priv_U des Benutzers durch Abrufen **475** und Entschlüsseln **470** der EPD unter Verwendung des wiedergewonnenen Kryptografieschlüssels K wiedergewinnen.

[0064] Wie bereits angegeben, können verschiedene Verifizierungsvorgehensweisen verwendet werden. Als Beispiel sei angenommen, dass der öffentliche Schlüssel Pub_U des Benutzers als Verifikationsdaten $v(i)$ verwendet wird. Dann kann der Wiedergewinnungsclient **220** Beweisdaten dadurch erzeugen **650**, dass er eine die verschiedenen Ad-hoc-Informationen $n(i)$ enthaltende Nachricht unter Verwendung des wiedergewonnenen Privatschlüssels Priv_U des Benutzers digital signiert. Jeder Verifikationsserver verifiziert **660** die erfolgreiche Wiedergewinnung der stark geheimen Daten K durch Verifizieren der digitalen Signatur unter Verwendung des öffentlichen Schlüssels Pub_U des Benutzers, und durch anschließendes Verifizieren, dass die korrekte Ad-hoc-Information $n(i)$ in der Nachricht enthalten ist. Andererseits sei angenommen, dass für die Verifikationsdaten $v(i) = h(K, \text{Id}(i))$ gilt. Dann können die Beweisdaten gemäß dem Ausdruck $g(v(i), n(i))$ berechnet werden, wobei g eine unidirektionale Funktion wie eine Kryptografie-Hashfunktion ist. Der Verifikationsserver verifiziert **660** die Beweisdaten durch Berechnen seines eigenen Werts aus seiner eigenen Kenntnis von $v(i)$ und $n(i)$ sowie durch Vergleichen des Ergebnisses mit dem empfangenen Wert.

[0065] Nachdem jeder Server die Beweisdaten empfangen hat, ermittelt er, ob die Zustandsvariable eine für die Ad-hoc-Information $n(i)$ anhängige Verifizierung anzeigt. Wenn eine Verifizierung anhängig ist, verifiziert der Server, dass die empfangenen Beweisdaten erfolgreich die Kenntnis der stark geheimen Daten K und die Frische in Verbindung mit der Ad-hoc-Information $n(i)$ demonstrieren. Wenn beide verifiziert sind, werden die Zähler für nicht verifizierte Wiedergewinnungsversuche für das Benutzerkonto U und das Passwort PWD dekrementiert. Andernfalls wird der Wiedergewinnungsprozess als nicht erfolgreich angesehen.

[0066] Die Verfahren **300**, **400**, **500** und **600** sind besonders vorteilhaft, da sie für viele Angriffstypen

resistent sind, einschließlich Angriffen durch die Server $S(i)$ oder auf diesen. Nachfolgend sind einige Angriffstypen und beispielhafte Gegenmaßnahmen angegeben.

[0067] Ein Angreifer kann versuchen, das Passwort PWD des Benutzers **110** zu erraten oder auf andere Weise zu gefährden. Dies kann dadurch bekämpft werden, dass der Wahl des Passworts PWD Erfordernisse auferlegt werden, um die Wahrscheinlichkeit zu verringern, dass ein Angreifer dasselbe erraten kann. Beispielsweise kann es erforderlich sein, dass das Passwort eine Minimallänge aufweist oder periodisch geändert wird. Bei einer anderen Gegenmaßnahme wird ein Versuchsschwellenwert ausgewählt, der die Anzahl der Rateversuche für das Passwort PWD beschränkt. So hat ein Angreifer nur eine eingeschränkte Anzahl von Versuchen zur Verfügung, um das Passwort PWD zu erraten. Wenn der Versuchsschwellenwert überschritten wird, wird der Erzeugungsprozess **500** vorzugsweise neu ausgeführt, wobei ein neues Passwort PWD und die Erzeugung neuer stark geheimer Daten K erforderlich sind, um so den Plan des Angreifers zu durchkreuzen. Der Versuchsschwellenwert sollte vorzugsweise so eingestellt werden, dass, angesichts der dem Passwort auferlegten Erfordernisse, die Wahrscheinlichkeit eines erfolgreichen Errateangriffs akzeptierbar klein ist, während immer noch ein Kompromiss hinsichtlich der Realität besteht, dass einige nicht erfolgreiche Versuche legitim sein können anstatt das Nebenprodukt eines Angriffs zu sein. Beispielsweise kann ein nicht erfolgreicher Versuch beim Fehlen jeglichen Angriffs als legitim aufgezeichnet werden, wenn der Benutzer **110** sein Passwort PWD falsch eintippt, und/oder als Ergebnis von Kommunikationsvorgängen oder eines Systemfehlers.

[0068] Alternativ kann ein Angreifer versuchen, das Protokoll dadurch zu umgehen, dass er w , a , und/oder $b(i)$ aus den zwischen den verschiedenen Komponenten übertragenen Nachrichten berechnet. Jedoch kann dieser Angriff dadurch abgewehrt werden, dass die Gruppe G so ausgewählt wird, dass sie gegen Attacken mit diskretem Logarithmus resistent ist. Wenn beispielsweise G die Reihe ganzer Zahlen modulo a zur Primzahl p ist, wird p entsprechend gut bekannten Regeln so ausgewählt, dass es sich um einen ausreichend starken Diffie-Hellman-Modul handelt (beispielsweise eine sichere Primzahl). Dann sind derartige Angriffe wegen des Problems mit diskretem Logarithmus nicht ausführbar. Dies gilt auch für einen angreifenden Server. Es ist für einen Server nicht ausführbar, andere Werte von w , a und/oder $b(i)$ als seine eigenen zu berechnen. Die Auswahl einer starken Primzahl p führt auch zu stark geheimen Daten K , was es als unausführbar macht, Chiffriertext, der gemäß den stark geheimen Daten K verschlüsselt wurde, direkt anzugreifen.

[0069] Bei einem anderen Angriffstyp kann der Angreifer als Wiedergewinnungsclient auftreten und schwache Serveranfragedaten M_1 an den Server $S(i)$ senden, wobei die Absicht vorliegt, gewisse Information zu den Servergeheimdaten $b(i)$ zu erhalten, wenn der Server die Serverantwortdaten $c(i) = M_1^{b(i)}$ zurück liefert. Der bekannte Hauptangriff dieses Typs stützt sich auf einen Kleingruppenangriff in Diffie-Hellman-Kryptographiesystemen. Dies kann durch verschiedene Maßnahmen abgewehrt werden, einschließlich einer anfänglichen Auswahl der Primzahl p mit geeigneten Eigenschaften, wie einer sicheren Primzahl der Form $p = 2q + 1$, wobei q eine Primzahl ist, und dadurch, dass dafür gesorgt wird, dass $b(i)$ immer eine gerade Zahl ist. Abhängig von den Eigenschaften des speziellen Typs der Gruppe G kann der Server $S(i)$ auch die Stärke der empfangenen Daten M_1 prüfen und die Antwort zurückweisen, wenn er sie als schwachen Wert erkennt.

[0070] Ein Angreifer kann versuchen, eine Passwortänderung für ein Benutzerkonto oder mehrere durch einen Server $S(i)$ unterstützte Konten dadurch auszuschließen oder zu erzwingen, dass er entweder wiederholt falsche Serverantwortdaten M_1 sendet oder Protokolle stört, beispielsweise durch Abfangen und Verwerfen von Nachrichten, die Serverantwortdaten oder Beweisdaten transportieren. Das Ziel dieses Angriffs besteht darin, dafür zu sorgen, dass der Server ein Benutzerkonto oder mehrere sperrt. Die Auswirkung dieses Angriffs kann dadurch verringert werden, dass dafür gesorgt wird, dass Server $S(i)$ die Verarbeitung wiederholter M_1 -Anfragen für ein Benutzerkonto "drosseln". D.h., dass der Server $S(i)$ die Verarbeitung für ein Benutzerkonto aufhebt oder verzögert, wenn während einer bestimmten Zeitperiode mehrere nicht erfolgreiche Versuche vorliegen.

[0071] Alternativ kann ein Angreifer versuchen, einen Server $S(i)$ dadurch zu überlasten, dass er eine massive Anzahl falscher Daten M_1 an den Server sendet, um so dafür zu sorgen, dass dieser eine massive Anzahl nutzloser aber rechenmäßig umfangreicher Potenzierungen ausführt. Wenn sich derartige Versuche gegen dasselbe Benutzerkonto richten, bekämpft ein Drosseln (wie oben erörtert) diesen Angriff wesentlich. So ist dieser Angriff hauptsächlich dann realisierbar, wenn eine große Anzahl verschiedener Benutzerkonten verwendet wird. Demgemäß kann dem dadurch entgegengewirkt werden, dass es erschwert wird, eine gültige Benutzerkontokennung U zu erraten, beispielsweise dadurch, dass vermieden wird, dass Benutzerkontokennungen aus einer einfachen Zahlenabfolge hergeleitet werden.

[0072] Gemäß einem abschließenden Beispiel kann ein Kommunikations- oder Systemfehler an einem Server zu mehreren Fehlern im Wiedergewinnungsprotokoll an den anderen Servern führen, was zu einer unnötigen Sperre und einer erzwungenen Pass-

wortänderung für ein Benutzerkonto oder mehrere führt. Um dieses Problem zu verringern, sollten Server vorzugsweise für hohe Verfügbarkeit konzipiert werden, mit geeigneter Redundanz von Systemen und Kommunikationspfaden. Außerdem können, wenn ein Server unvermeidlich ausfällt, Verwaltungskontrollen den Betrieb der anderen Server vorübergehend aufheben. Ein Drosseln (wie oben erörtert) verringert ebenfalls das Auftreten einer Kontosperrung.

[0073] Die obige Beschreibung ist enthalten, um die Funktion der bevorzugten Ausführungsformen zu veranschaulichen, und sie hat keine Bedeutung dahingehend, den Schutzbereich der Erfindung einzuschränken. Aus der obigen Erörterung sind dem Fachmann viele Variationen ersichtlich, die vom Schutzbereich der Erfindung umfasst sind. Beispielsweise kann das Vorhandensein einer Hardwarekarte erforderlich sein, um den Wiedergewinnungsprozess **900** abzuschließen. Bei einer Vorgehensweise ermittelt der Wiedergewinnungsclient **220**, ob eine Hardwarekarte des Benutzers vorhanden ist, und wenn dies der Fall ist, überträgt er anschließend Daten, die diese Tatsache bestätigen oder beweisen, an jeden Geheimdaten speichernden Server. Die Server wiederum nehmen am Wiedergewinnungsprozess nur dann teil, wenn sie diese Daten vom Wiedergewinnungsclient **220** empfangen. Wenn eine Verwendung in Zusammenhang mit einer Hardwarekarte erfolgt, wie einer Aufforderungs-Antwort-Karte oder einem Zeitsynchronisierten Generator für ein einmaliges Passwort, auf solche Weise, dass der Wiedergewinnungsclient **220** zusätzlich den Besitz der Karte einem oder mehreren Servern gegenüber beweist, kann die sich ergebende Kombination von Verfahren effektiv anstelle eines Smartcard-Verfahrens arbeiten, bei dem stark geheime Daten aus der Smartcard wiedergewonnen werden, nachdem schwach geheime Daten des Benutzers, wie eine PIN, eingegeben wurden. Der Vorteil der ersteren Vorgehensweise gegenüber der Smartcard-Vorgehensweise ist eine mögliche wesentliche Verringerung der Aufstell- und Wartungskosten, da die o.g. Kartentypen keine speziellen Hardwareschnittstellen, wie sie bei Smartcards erforderlich sind, benötigen. Demgemäß ist der Schutzbereich der Erfindung nur durch die folgenden Ansprüche eingeschränkt.

Patentansprüche

1. Verfahren, um Vorrichtungen in die Lage zu versetzen, stark geheime Daten eines Benutzers sicher aus schwach geheimen Daten für den Benutzer wiederzugewinnen, wobei in den Verfahren: die schwach geheimen Daten des Benutzers bestimmt (**320**) werden; Serveranfragedaten für wenigstens einen ein Geheimnis speichernden Server (**130**) berechnet werden, wobei die Serveranfragedaten eine Funktion der schwach geheimen Daten und eines kurzlebigen Cli-

ent-Geheimnisses sind, und die Serveranfragedaten ohne Kenntnis des kurzlebigen Client-Geheimnisses keine Information über die schwach geheimen Daten preisgeben;

Serverantwortdaten von dem ein Geheimnis speichernden Server empfangen (**340**) werden, wobei die Serverantwortdaten eine Funktion der Server-Geheimnisdaten für den ein Geheimnis speichernden Server und der Serveranfragedaten sind, und die Serverantwortdaten ohne Kenntnis der schwach geheimen Daten und des kurzlebigen Client-Geheimnisses keine Information über die Server-Geheimnisdaten preisgeben;

eine geheime Komponente für den ein Geheimnis speichernden Server als eine Funktion der von dem ein Geheimnis speichernden Server empfangenen Serverantwortdaten und des kurzlebigen Client-Geheimnisses berechnet (**330**) wird, wobei die geheime Komponente eine Funktion der schwach geheimen Daten und des Servergeheimnisses aber unabhängig von dem kurzlebigen Geheimnis ist;

die stark geheimen Daten des Benutzers als Funktion der geheimen Komponente berechnet (**330**) werden; und

Verifikationsdaten für wenigstens einen Verifikations-Server (**130J-130N**) bestimmt (**350**) werden, wobei die Verifikationsdaten den Verifikations-Server dazu in die Lage versetzen zu verifizieren, dass eine Vorrichtung die stark geheimen Daten erfolgreich wiedergewonnen hat, aber es für den Verifikations-Server rechnerisch undurchführbar ist, die schwach geheimen Daten nur aufgrund eines Zugriffs auf seine Verifikationsdaten zu bestimmen.

2. Verfahren nach Anspruch 1, wobei die Server-Geheimnisdaten eine ganzzahlige Zufallszahl b aufweisen;

die Geheimniskomponente ein Wert $K = h(w^b)$ ist, wobei

h eine Funktion ist, und

$w = f(\text{schwach geheime Daten})$, wobei f eine Funktion ist, die ein Element einer finiten Gruppe G erzeugt, in der eine Potenzierung effizient ist, aber das abstrakte logarithmische Problem rechnerisch undurchführbar ist;

die Potenzierung w^b in der Gruppe G berechnet wird; das kurzlebige Client-Geheimnis eine ganzzahlige Zufallszahl a ist, für die es eine entsprechende ganzzahlige Zahl a' gibt, so dass $x^{aa'} = x$ für alle x in Gruppe G gilt;

die Serveranfragedaten als der Wert $M = w^a$ berechnet werden, wobei die Potenzierung in der Gruppe G berechnet wird;

die Serverantwortdaten als der Wert $c = M^b$ berechnet werden, wobei die Potenzierung in der Gruppe G berechnet wird; und

die geheime Komponente als der Wert $K = h(c^{a'})$ berechnet wird, wobei die Potenzierung in der Gruppe G berechnet wird.

3. Verfahren nach Anspruch 2, wobei die Gruppe G gewählt ist aus:

einer multiplikativen Gruppe der Reihe von ganzen Zahlen modulo p , wobei p eine große Primzahl ist, die als ein Diffie-Hellman-Modul geeignet ist; und einer Gruppe von Punkten einer elliptischen Kurve über ein begrenztes Feld.

4. Verfahren nach Anspruch 1, wobei ferner private Daten für den Benutzer unter Verwendung der stark geheimen Daten als Kryptographieschlüssel in einem symmetrischen Kryptographiesystem verschlüsselt werden, und

beim Bestimmen der Verifikationsdaten öffentliche Daten bestimmt werden, die den privaten Daten des Benutzers entsprechen, wobei eine erste Einheit mit Zugriff auf die privaten Daten den Zugriff auf eine zweite Einheit mit Zugriff auf die öffentlichen Daten nachweisen kann, ohne der zweiten Einheit die privaten Daten zu offenbaren.

5. Verfahren nach Anspruch 1, wobei beim Bestimmen der Verifikationsdaten Verifikationsdaten als Einbahnstraßenfunktion der stark geheimen Daten berechnet werden.

6. System (**100**), um Vorrichtungen in die Lage zu versetzen, stark geheime Daten eines Benutzers aus schwach geheimen Daten für den Benutzer wiederzugewinnen, wobei

das System einen Erzeugungs-Client (**120**) und wenigstens einen ein Geheimnis speichernden Server (**130**) aufweist, der an den Erzeugungs-Client angeschlossen ist, und

der Client und der Server Einrichtungen zum Ausführen von Schritten aufweisen, in denen:

der Erzeugungs-Client die schwach geheimen Daten des Benutzers bestimmt;

der Erzeugungs-Client Serveranfragedaten berechnet und an den ein Geheimnis speichernden Server übermittelt, wobei die Serveranfragedaten eine Funktion der schwach geheimen Daten und eines kurzlebigen Client-Geheimnisses sind, und die Serveranfragedaten ohne Kenntnis des kurzlebigen Client-Geheimnisses keine Information über die schwach geheimen Daten preisgeben;

der ein Geheimnis speichernde Server Serverantwortdaten berechnet und an den Erzeugungs-Client übermittelt, wobei die Serverantwortdaten eine Funktion der Server-Geheimnisdaten für den ein Geheimnis speichernden Server und der Serveranfragedaten sind, und die Serverantwortdaten ohne Kenntnis der schwach geheimen Daten und des kurzlebigen Client-Geheimnisses keine Information über die Server-Geheimnisdaten preisgeben;

der Erzeugungs-Client eine geheime Komponente für den ein Geheimnis speichernden Server als eine Funktion der von dem ein Geheimnis speichernden Server empfangenen Serverantwortdaten und des kurzlebigen Client-Geheimnisses berechnet, wobei

die geheime Komponente eine Funktion der schwach geheimen Daten und des Servergeheimnisses aber unabhängig von dem kurzlebigen Geheimnis ist; der Erzeugungs-Client die stark geheimen Daten des Benutzers als Funktion der geheimen Komponente berechnet; und der Erzeugungs-Client Verifikationsdaten für wenigstens einen Verifikations-Server (**130J-130N**) bestimmt, wobei die Verifikationsdaten den Verifikations-Server dazu in die Lage versetzen zu verifizieren, dass eine Vorrichtung die stark geheimen Daten erfolgreich wiedergewonnen hat, aber es für den Verifikations-Server rechnerisch undurchführbar ist, die schwach geheimen Daten nur aufgrund eines Zugriffs auf seine Verifikationsdaten zu bestimmen.

7. Verfahren nach Anspruch 6, wobei die Server-Geheimnisdaten eine ganzzahlige Zufallszahl b aufweisen; die Geheimniskomponente ein Wert $K = h(w^b)$ ist, wobei h eine Funktion ist, und $w = f(\text{schwach geheime Daten})$, wobei f eine Funktion ist, die ein Element einer finiten Gruppe G erzeugt, in der eine Potenzierung effizient ist, aber das abstrakte logarithmische Problem rechnerisch undurchführbar ist; die Potenzierung w^b in der Gruppe G berechnet wird; das kurzlebige Client-Geheimnis eine ganzzahlige Zufallszahl a ist, für die es eine entsprechende ganzzahlige Zahl a' gibt, so dass $x^{aa'} = x$ für alle x in Gruppe G gilt; die Serveranfragedaten als der Wert $M = w^a$ berechnet werden, wobei die Potenzierung in der Gruppe G berechnet wird; die Serverantwortdaten als der Wert $c = M^b$ berechnet werden, wobei die Potenzierung in der Gruppe G berechnet wird; und die geheime Komponente als der Wert $K = h(c^a)$ berechnet wird, wobei die Potenzierung in der Gruppe G berechnet wird.

8. Verfahren nach Anspruch 7, wobei die Gruppe G gewählt ist aus: einer multiplikativen Gruppe der Reihe von ganzen Zahlen Modulo p , wobei p eine große Primzahl ist, die als ein Diffie-Hellman-Modul geeignet ist; und einer Gruppe von Punkten einer elliptischen Kurve über ein begrenztes Feld.

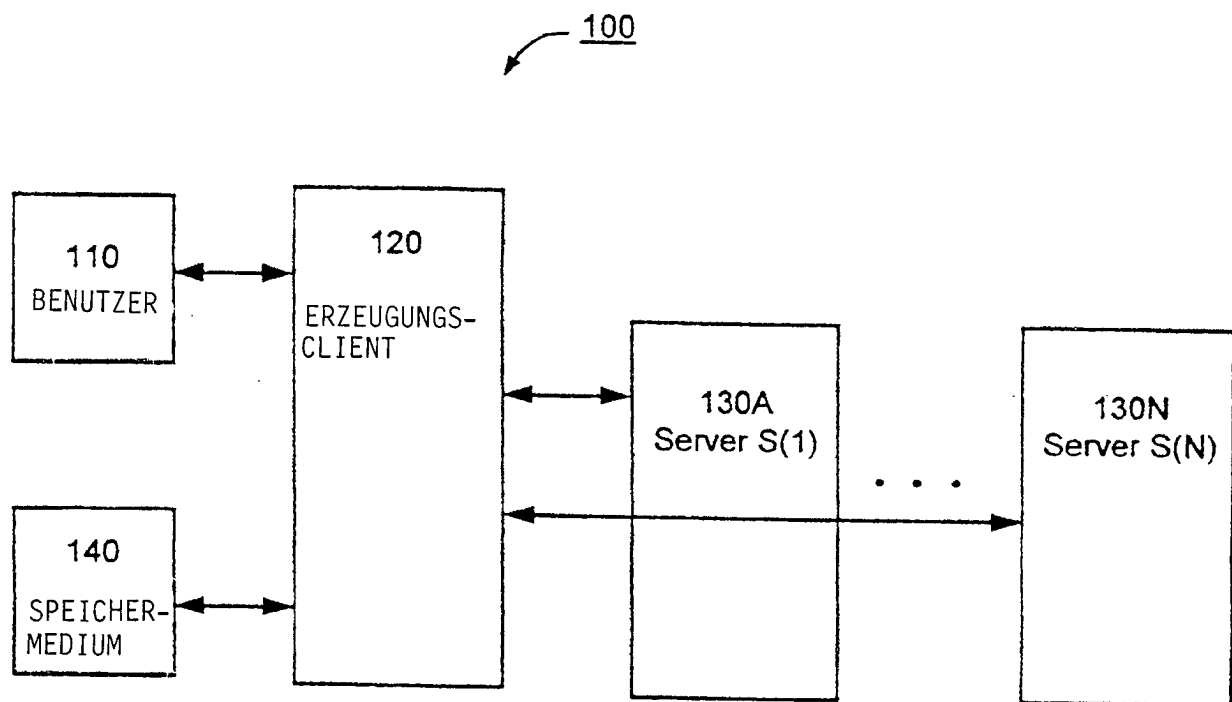
9. Verfahren nach Anspruch 6, wobei ferner der Erzeugungs-Client private Daten für den Benutzer unter Verwendung der stark geheimen Daten als Kryptographieschlüssel in einem symmetrischen Kryptographiesystem verschlüsselt, und beim Bestimmen der Verifikationsdaten öffentliche Daten bestimmt werden, die den privaten Daten des Benutzers entsprechen, wobei eine erste Einheit mit Zugriff auf die privaten Daten den Zugriff auf eine

zweite Einheit mit Zugriff auf die öffentlichen Daten nachweisen kann, ohne der zweiten Einheit die privaten Daten zu offenbaren.

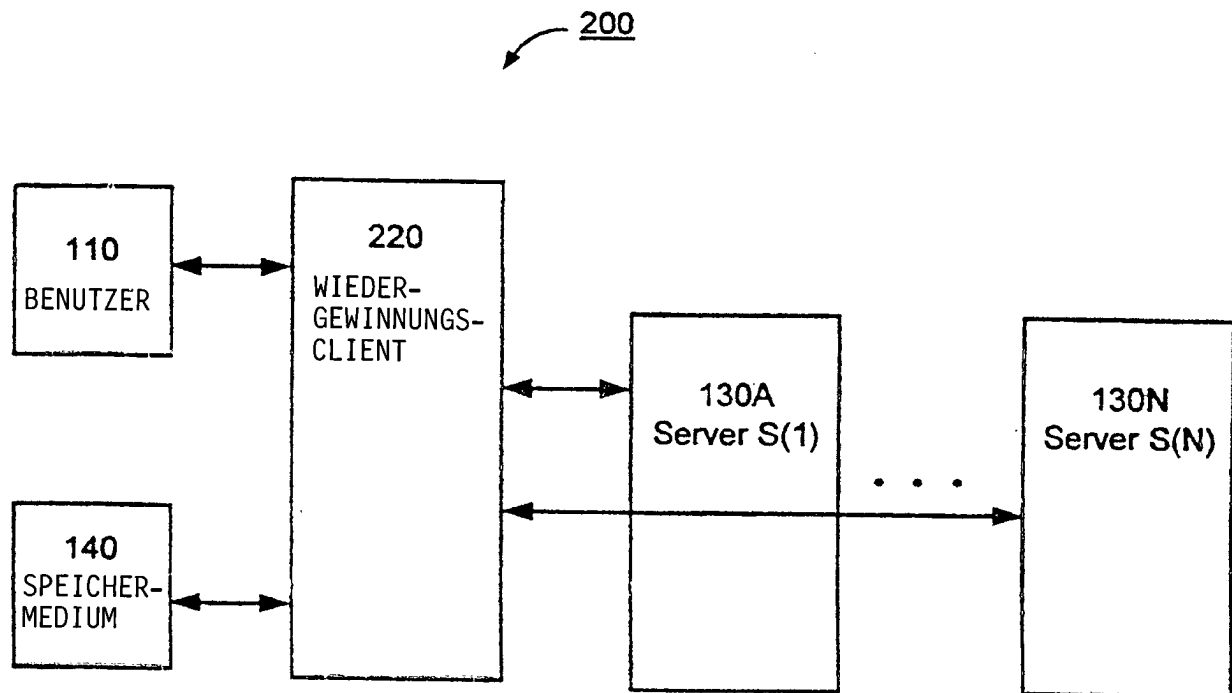
10. Verfahren nach Anspruch 6, wobei beim Bestimmen der Verifikationsdaten Verifikationsdaten als Einbahnstraßenfunktion der stark geheimen Daten berechnet werden.

11. Computerprogrammprodukt mit durch einen Erzeugungs-Clientcomputer ausführbaren Befehlen, um den Erzeugungs-Clientcomputer anzuweisen, alle Schritte eines Verfahrens gemäß einem der Ansprüche 1 bis 5 auszuführen.

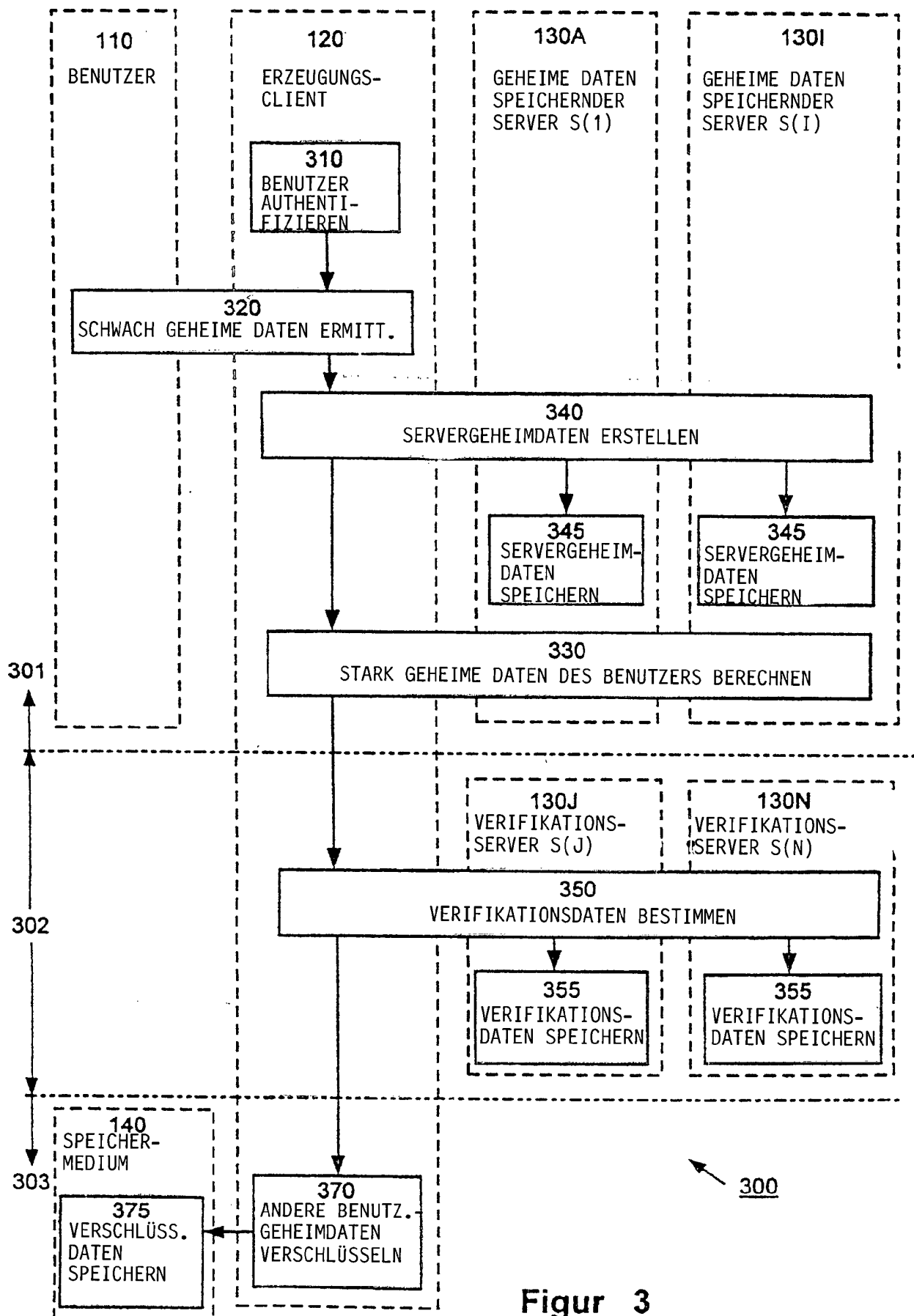
Es folgen 6 Blatt Zeichnungen



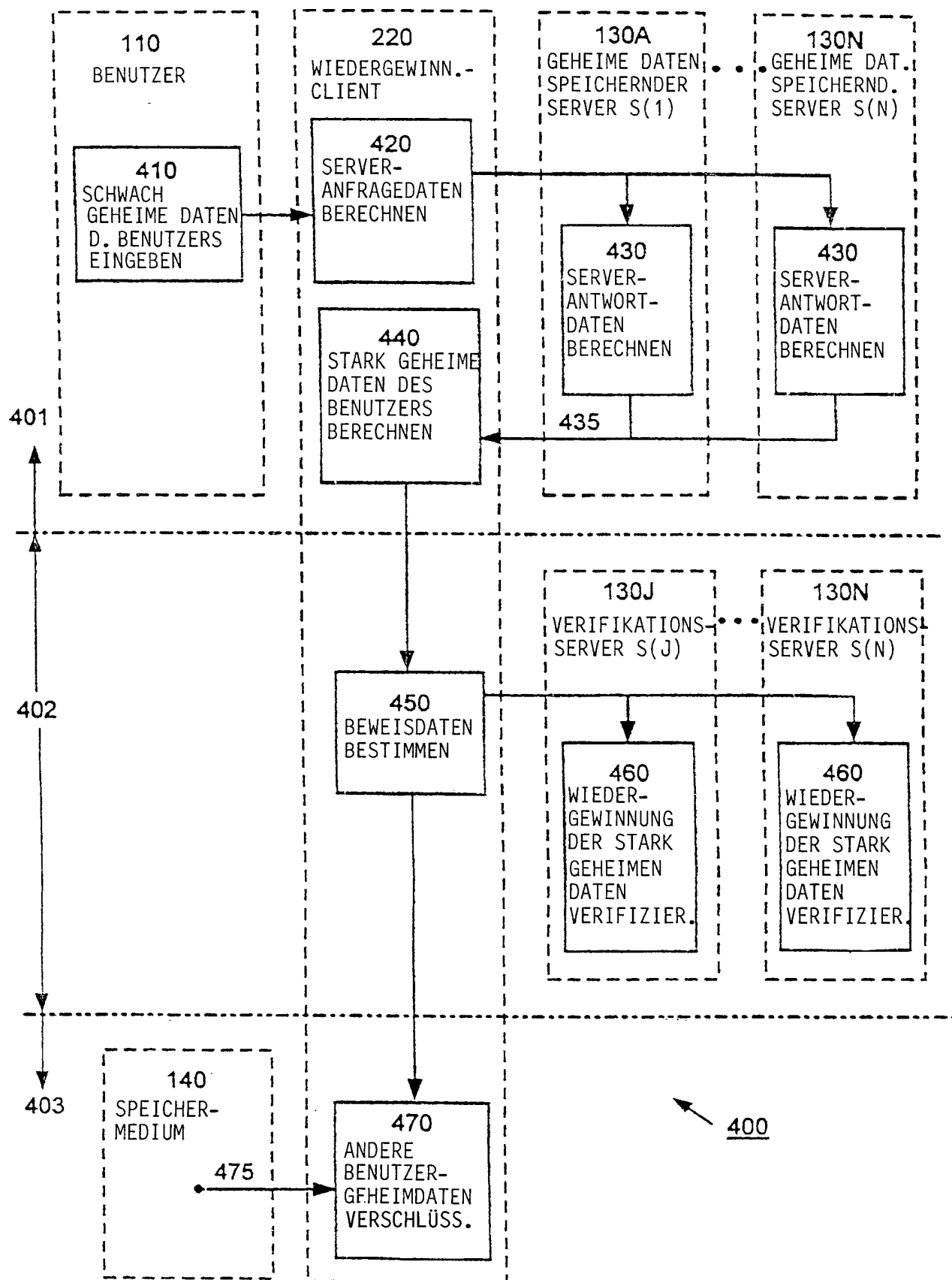
Figur 1



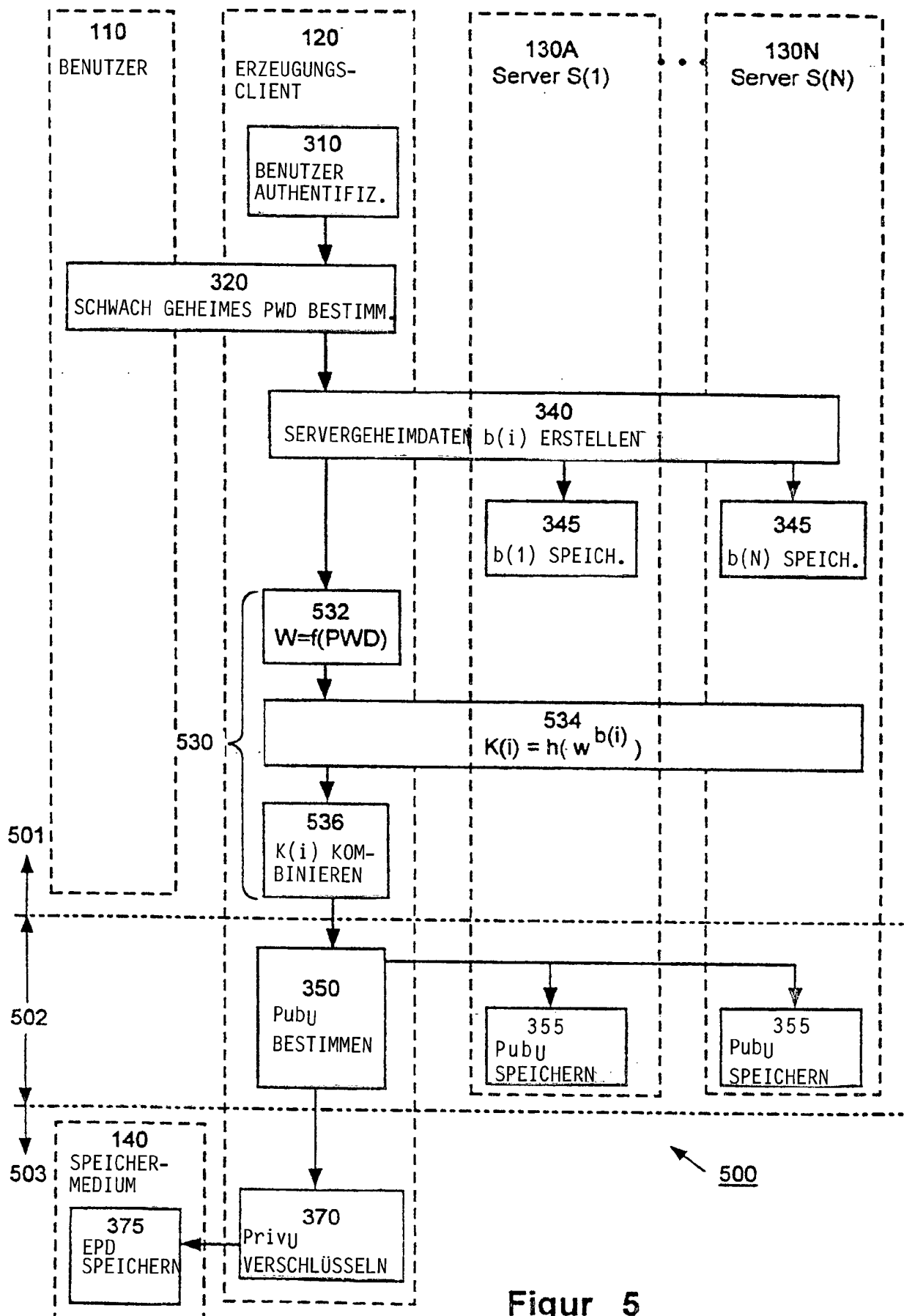
Figur 2



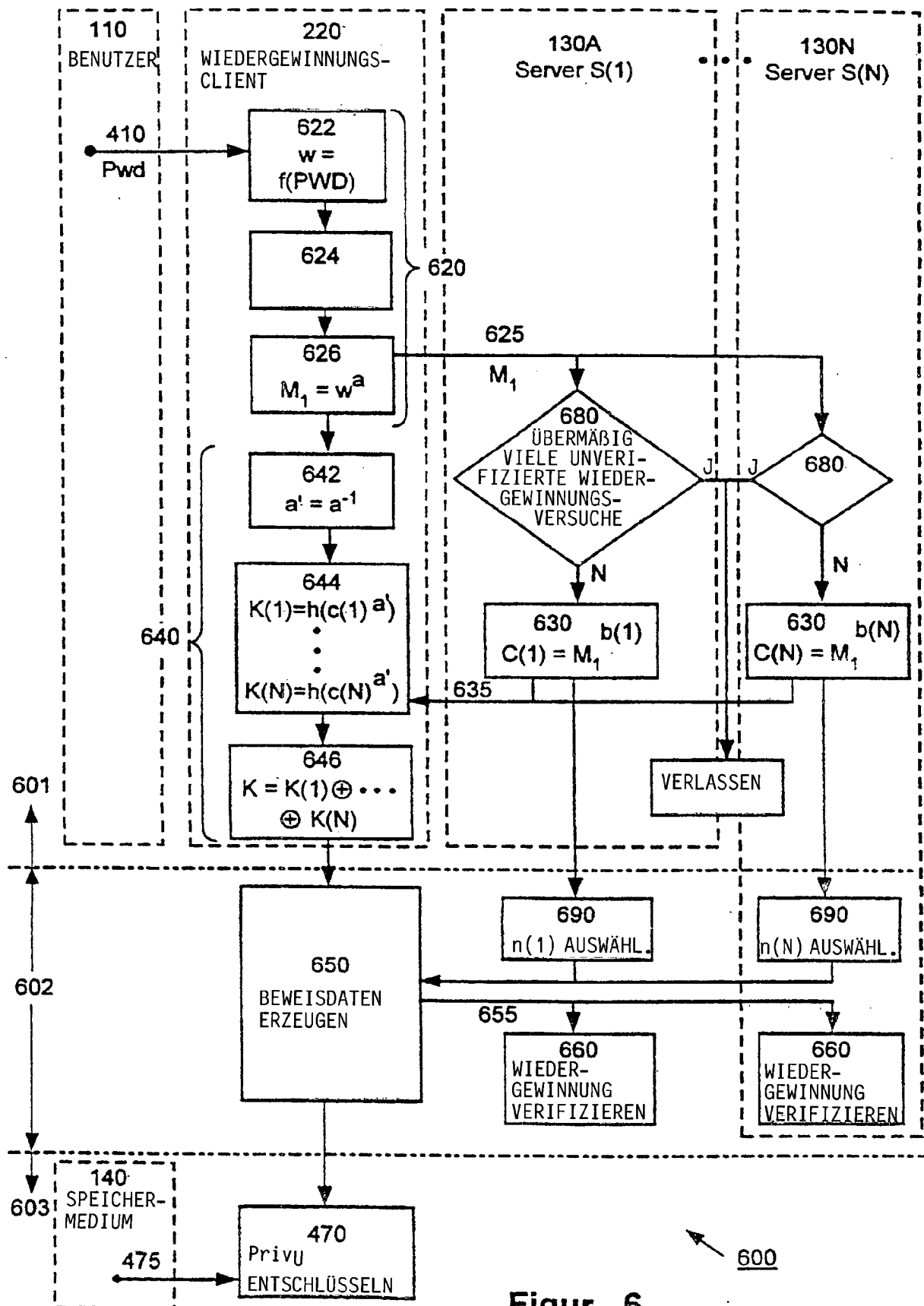
Figur 3



Figur 4



Figur 5



Figur 6