(19) Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 102 48 542 A1** 2004.04.22

(12) # Offenlegungsschrift

(21) Aktenzeichen: **102 48 542.9**
(22) Anmeldetag: **14.10.2002**
(43) Offenlegungstag: **22.04.2004**

(51) Int Cl.[7]: **G06F 12/14**

(71) Anmelder:
Deutsche Telekom AG, 53113 Bonn, DE

(72) Erfinder:
Schwenk, Jörg, 91239 Henfenfeld, DE

(56) Für die Beurteilung der Patentfähigkeit in Betracht zu ziehende Druckschriften:
US 59 56 404

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Rechercheantrag gemäß § 43 Abs. 1 Satz 1 PatG ist gestellt.

(54) Bezeichnung: **Verfahren zur Sicherung von Log-Files**

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zur Sicherung von Log-Files mittels einer Datenverarbeitungsanlage gegen unerlaubte Veränderung, bei dem zu einem Eintrag im Log-File ein zweiter Eintrag generiert wird, bei dem die Datenverarbeitungsanlage ein Sicherheitsmodul umfasst, welches zu jedem Eintrag im Log-File eine, insbesondere fortlaufende, Information bereitstellt und wobei jedem Eintrag ein Funktionswert, insbesondere eine digitale Signatur, zugeordnet wird, der wenigstens aus der Information und dem Eintrag ermittelt wird.

Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zur Sicherung von Logfiles mittels einer Datenverarbeitungsanlage gegen unerlaubte Veränderung, bei dem zu einem Eintrag im Logfile ein zweiter Eintrag generiert wird.

[0002] Logfiles sind bekannt, um z.B. erlaubte Zugriffe und unerlaubte Manipulationen an den Daten eines Computers bzw. an dem Computer selbst unterscheiden zu können. In diesen Logfiles werden bestimmte Aktionen an den Daten bzw. am Computer protokolliert, um z.B. einen Fehler- oder auch Angriffsfall erkennen und die Ursache eingrenzen zu können. Diese Logfiles selbst sind in der Regel jedoch nicht gegen Manipulationen geschützt.

[0003] Es ist weiterhin bekannt Logfiles durch den Einsatz z.B. kryptographischer Hashfunktionen zu schützen. Eine Besonderheit stellt die sogenannte Einweg-Hashfunktion dar, bei der aus den ermittelten Hashwerten nicht auf die Eingaben zurückgeschlossen werden kann. Solche Hash-Funktionen müssen kollisionsfrei sein, was bedeutet, dass keine zwei unterschiedlichen Eingaben gefunden werden können (manuell oder maschinell), aus denen derselbe Hash-Wert generiert wird (obwohl es solche Paare natürlich vielfach gibt).

[0004] Ein besonders einfaches Beispiel einer Hashfunktion stellt das Register eines Adressbuches dar, bei dem der endlichen Menge der Buchstaben des Alphabetes unendlich viele Namenseinträge zurgeordnet werden können. Diese einfache Hash-Funktion ist jedoch nicht kollisionsfrei.

[0005] Allgemein gilt, dass eine Einweg-Hashfunktion H einer beliebig langen Nachricht N einen Hashwert h = H(N) zuordnet, der z.B. immer feste Länge aufweist, z.B. 128, 160, 192 oder 256 Bit. Wie erwähnt ist es schwierig bis unmöglich, aus einem gegebenen Hashwert h auf die Nachricht N zurückzuschliessen.

[0006] So besteht in einer bekannten Anwendung die Möglichkeit auf eine übermittelte Nachricht N die Hashfunktion H anzuwenden und den Hashwert h2 = H(N) mit einem parallel übermittelten Hashwert h1 zu vergleichen, um so festzustellen, ob die Nachricht auf dem Weg der Übermittlung manipuliert wurde. Eine Manipulationsfreiheit besteht, wenn h2 = h1, also die Hashwerte übereinstimmen.

[0007] Auf diese Weise können z.B. auch die angesprochenen Logfiles geschützt werden, indem zu jedem Dateneintrag im Logfile ein zugehöriger Hashwert berechnet wird. Um diese Methode weiterhin zu sichern ist es bekannt, zu jedem neuen Eintrag im Logfile einen Hashwert zu berechnen, der auf dem neuen Dateneintrag und dem bisherigen Hashwert beruht. Der aktuelle Hashwert ist somit eine Kurzfassung des gesamten bisherigen Logfiles. Diese Methode ist auch als das „ewige Logfile" bekannt.

[0008] Ein typisches Logfile dieser Art kann z.B. wie folgt aussehen:

Data1	h1 = H(Data1)
Data2	h2 = H(h1, Data1)
Data3	h3 = H(h2, Data2) usw. wenn H die Hashfunktion darstellt.

[0009] Hashfunktionen als solche sind dem Fachmann allgemein bekannt und bedürfen keiner weiteren Erklärung. Bekannte Algorithmen sind z.B. MD5, der von Ronal L. Rivest entwickelt wurde, der SHA-1 (Secure Hash Algorithm) der National Security Agency (NSA) der USA und der sogenannte Tiger-Algorithmus von Eli Biham und Ross Anderson.

[0010] Eine weitere Sicherung von Hashfunktionen bietet die Verwendung von Schlüsseln, was bedeutet, dass ein durch eine solche schlüsselabhängige Hashfunktion errechneter Hashwert nur dann verifiziert werden kann, wenn der verifizierenden Person der Schlüssel zu der Hashfunktion bekannt ist. Solche schlüsselabhängige Hashfunktionen werden auch als MAC (Message Authentication Code) bezeichnet.

[0011] Ebenfalls bekannt ist es Daten und/oder Nachrichten durch eine digitale Signatur zu schützen, bzw. durch eine solche die Möglichkeit zu eröffnen, feststellen zu können, ob eine Nachricht manipuliert wurde.

[0012] Eine digitale Signatur bildet somit eine Art Fingerabruck der Daten oder der Nachricht bzw. eine Art Unversehrtheits-Siegel. Typisch für eine digitale Signatur ist ein Schlüsselpaar bestehend aus einem privaten und einem öffentlichen Schlüssel. Die Signatur einer Nachricht oder von Daten wird mittels des kryptographischen privaten Schlüssels erzeugt um mittels des öffentlichen Schlüssels vom Empfänger der Daten oder der Nachricht die Unversehrtheit zu prüfen. So kann der Empfänger prüfen, ob eine Manipulation stattgefunden hat.

[0013] Da die Anwendung eines Algorithmus zu Erstellung einer digitalen Signatur vergleichbar langsam erfolgt ist es bekannt die Daten zunächst durch eine Hashfunktion zu komprimieren und dann auf dieses Komprimat die digitale Signatur anzuwenden.

[0014] Nachteilig bei der Anwendung einer Hashfunktion auf die neu hinzugekommenen Daten in einem Logfile, z.B. in einem ewigen Logfile ist es, dass diese oben beschriebene Methode keinen Schutz dagegen bietet, die z.B. letzten n Einträge aus dem Logfile zu löschen und gegen geänderte m neue Einträge zu ersetzen, wobei die manipulierende Person oder ein Computervirus die neuen Hashwerte berechnen kann, so dass diese Manipulation nicht auffällt. Weiterhin nachteilig ist es, dass beim ewigen Logfile alle jemals gehashten Daten auf unbestimmte Zeit gespeichert werden müssen. Geht auch nur einer dieser Datensätze verloren, so verliert

das ewige Logfile seine Beweiskraft.

[0015] Aufgabe der Erfindung ist es, einen verbesserten Schutz von Logfiles zur Verfügung zu stellen um so eine Manipulation erkennbar zu machen.

[0016] Diese Aufgabe wird erfindungsgemäß dadurch gelöst, dass die Datenverarbeitungsanlage ein Sicherheitsmodul umfasst, welches zu jedem Eintrag im Log-File eine, insbesondere fortlaufende, Information bereitstellt und wobei jedem Eintrag ein Funktionswert, insbesondere eine digitale Signatur zugeordnet wird, der wenigstens aus der Information und dem Eintrag ermittelt wird.

[0017] Dieser Funktionswert kann gebildet werden durch eine beliebige mathematische Funktion, bevorzugt durch eine digitale Signatur, eine Hashfunktion, insbesondere eine Einweg-Hashfunktion, oder ein MAC.

[0018] Die Verwendung einer solchen zur Verfügung gestellten Information hat den Vorteil, dass eine Manipulation auffällt, da einem Computervirus oder einer manipulierenden Person z.B. bei der Entfernung und Ersetzung von Daten nicht bekannt ist, mit welcher zusätzlichen Information zu den neuen Daten der Algorithmus angewendet werden muss.

[0019] Besonders einfach ist das erfindungsgemäße Verfahren, wenn die Information ein fortlaufender Zahlenwert ist.

[0020] Vorteilhaft ist es weiterhin, wenn das Sicherheitsmodul, welches z.B. als eine Chipkarte ausgebildet ist, sowohl die zusätzliche Information bereitstellt, als auch die Anwendung des Algorithmus bzw. der Funktion, insbesondere eine digitale Signatur oder MAC durchführt. Bei der Chipkarte erfolgt eine Realisierung z.B. in Übereinstimmung mit DIN-17.4, welche die Spezifikationen gemäß SigG und SigV festlegt.

[0021] Zur Beschleunigung der Funktionswertbildung und zur weiteren Sicherung kann es bevorzugt vorgesehen sein, dass vor der Ermittlung dieses Funktionswertes, insbesondere der Signatur, ein erster Algorithmus, insbesondere eine Hashfunktion, auf wenigstens den Eintrag zur Ermittlung eines ersten Wertes angewendet wird und der Ermittlung des Funktionswertes, insbesondere der Signatur, wenigstens die Information und der berechnete erste Wert zugrunde gelegt wird.

[0022] Dieser erste Wert kann gebildet werden durch einen beliebigen mathematischen Algorithmus, bevorzugt durch eine Hashfunktion, insbesondere eine Einweg-Hashfunktion, eine digitale Signatur, oder ein MAC.

[0023] In einer besonderen Ausgestaltung des Verfahrens kann es weiterhin vorgesehen sein, dass der erste Wert zu einem Eintrag und dem ersten Wert eines vorherigen Eintrages, insbesondere sämtlicher vorheriger Einträge und/oder ersten Werte berechnet wird. Dies entspricht wiederum dem Fortschreiben des Logfiles im Sinne des vorgenannten „ewigen Logfiles".

[0024] Ein konkretes Ausführungsbeispiel der Erfindung wird nachfolgend beschrieben.

[0025] Gemäß der grundlegenden Idee des erfindungsgemäßen Verfahrens wird in einem Logfile zu jedem neuen fortlaufenden Dateneintrag Data an n-ter Stelle des Logfiles ein Funktionswert fn aufgrund der Funktion F gebildet, der das Logfile ergänzt. Zur Bildung dieses Funktionswertes fn stellt das Sicherheitsmodul eine Information In zur Verfügung. Ein Logfile kann dementsprechend den folgenden Aufbau haben:

Data1 f1 = F(I1, Data1)
Data2 f2 = F(I2, Data2)
Datan fn = F(In, Datan)

[0026] In der bevorzugten Form kann vor der Anwendung der Funktion F zur Bildung eines ersten Wertes ein erster Algorithmus A angewendet werden, so dass sich folgender Aufbau des Logfiles ergibt:

Data1 f1 = F(I1, A(Data1))
Data2 f2 = F(I2, A(Data2))
Datan fn = F(In, A(Datan))

[0027] Zur Fortschreibung des Logfiles (im Sinne eines ewigen Logfiles) kann auch hier der Algorithmus A nicht nur auf den jeweiligen neuen Eintrag Data, sondern auch auf die vorherigen berechneten ersten Werte An-1 und/oder die vorherigen Funktionswerte fn-1 angewendet werden. Hierdurch ergibt sich z.B. folgender Aufbau des Logfiles:

Data1 f1 = F(I1, A(Data1))
Data2 f2 = F(I2, A(A1, Data2)) mit A1 = A(Data1)
Datan fn = F(In, A(An-1, Datan)) mit An-1 = A(Datan-1) oder:
Data1 f1 = F(I1,A(Data1))
Data2 f2 = F(I2, A(f1, Data2)) mit f1 = F(I1, A(Data1))
Datan fn = F(In, A(fn-1, Datan)) mit fn-1 = F(In-1, A(fn-1, Datan-1))

[0028] Wie vorangehend erwähnt kann für die Funktion F und für den Algorithmus A jeweils z.B. eine Hashfunktion, eine digitale Signatur oder ein MAC oder jede andere beliebige mathematische Funktion eingesetzt werden. Die von dem Sicherheitsmodul zur Verfügung gestellte Information I kann z.B. ein fortlaufender Zahlenwert 1,2,3,4,5.... sein, so dass sich in einer besonders bevorzugten Ausführungsform folgender Aufbau eines beispielsweise manipulierten Logfiles ergibt:

Data1 sig1 = sig(1, hash(Data1))
Data2 sig2 = sig(2, hash(Data2))

Data3 sig3 = sig(3, hash(Data3))
Data3+n sig3+n = sig(3+n, hash(Data3+n))
Data3+n+1 sig3+n+1 = sig(3+n+1, hash(Data3+n+1))

[0029] Bei einer Überprüfung von sig3+n würde sich ergeben, dass nicht (4, hash(Data3+n)) signiert wurde, sondern (3+n, hash(Data3+n)) und die Manipulation bzw. Lücke im Logfile wäre offensichtlich.

[0030] In einer bevorzugten Ausführungsform beinhaltet das Verfahren eine Kommunikation zwischen dem Computer, auf dem das Logfile gespeichert wird, und dem Sicherheitsmodul.

Beispiel:	**Data1,**	**sig1=sig(1, hash(IV,Data1))**
	Data2,	**sig2=sig(2, hash(sig1, Data2))**
	Data3,	**sig3=sig(3, hash(sig2, Data3))**

[0031] In diesem Beispiel werden die einzelnen Einträge des Logfiles dadurch verkettet, dass jeweils in die Signatur eines neuen Eintrags der Wert der Signatur für den vorangehenden Eintrag mit einfließt. (Der Wert IV steht hier im Sinne einheitlicher Datenformate für den fehlenden Wert sig0.) Um nun z.B. sig3 zu erzeugen, läuft folgende Interaktion zwischen Computer und Sicherheitsmodul ab:

1. Der Computer berechnet hash(sig2, Data3) und sendet diesen Wert an das Sicherheitsmodul. Der Computer kann diese Berechnung durchführen, da er beide Werte kennt. Das Sicherheitsmodul wird durch diese Vorgehensweise von der Verarbeitung größerer Datenmengen entlastet, da ein großer Datensatz Data3 z. B. von einer Chipkarte nur sehr langsam verarbeitet werden könnte.
2. Das Sicherheitsmodul berechnet die digitale Signatur sig3 aus dem erhaltenen Wert hash(sig2, Data3) und einem internen Zählerwert, die in diesem Beispiel auf „3" gesetzt wurde. Dazu benötigt das Sicherheitsmodul einen privaten Schlüssel, der außerhalb des Sicherheitsmoduls nicht existiert. Das Sicherheitsmodul überträgt anschließend den Wert sig3 an den Computer.
3. Der Computer speichert sig3 an der vorgegebenen Stelle im Logfile.

Patentansprüche

1. Verfahren zur Sicherung von Log-Files mittels einer Datenverarbeitungsanlage gegen unerlaubte Veränderung, bei dem zu einem Eintrag im Log-File ein zweiter Eintrag generiert wird, **dadurch gekennzeichnet**, dass die Datenverarbeitungsanlage ein Sicherheitsmodul umfasst, welches zu jedem Eintrag im Log-File eine, insbesondere fortlaufende, Information bereitstellt und wobei jedem Eintrag ein Funktionswert, insbesondere eine digitale Signatur, zugeordnet wird, der wenigstens aus der Information und dem Eintrag ermittelt wird.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Information ein fortlaufender Zahlenwert ist.

3. Verfahren nach einem der vorherigen Ansprüche, dadurch gekennzeichnet, dass vor der Ermittlung des Funktionswertes, insbesondere der Signatur, ein erster Algorithmus, insbesondere eine Hashfunktion, auf wenigstens den Eintrag zur Ermittlung eines ersten Wertes angewendet wird und der Ermittlung des Funktionswertes, insbesondere der Signatur, wenigstens die Information und der berechnete erste Wert zugrunde gelegt wird.

4. Verfahren nach Anspruch 3, dadurch gekennzeichnet, dass der erste Wert berechnet wird aus einem Eintrag und dem ersten Wert eines vorherigen Eintrages oder aus einem Eintrag und dem Funktionswert eines vorherigen Eintrages, insbesondere sämtlicher vorheriger Einträge und/oder ersten Werte.

5. Verfahren nach einem der vorherigen Ansprüche dadurch gekennzeichnet, dass die Funktion und/oder der Algorithmus zur Bildung des ersten Wertes als eine digitale Signatur und/oder Hash-Funktion und/oder als MAC ausgebildet ist.

Es folgt kein Blatt Zeichnungen