

Praktikabler Datenschutz für Log-Daten *

Ulrich Flegel
Universität Dortmund
D-44221 Dortmund
ulrich.flegel@udo.edu

Zusammenfassung

Dieser Beitrag zeigt die Schwierigkeit auf, Log-Daten im Einklang mit den geltenden Datenschutz-Gesetzen zu verwenden. Es wird die Pseudonymisierung der Log-Daten vorgeschlagen, so daß durch die technische Zweckbindung bei der Pseudonym-Aufdeckung die Interessen Anonymität und Zurechenbarkeit zu einem fairen Ausgleich kommen. Gleichzeitig sind die Log-Daten durch die Pseudonymisierung dem Geltungsbereich der Datenschutz-Gesetze entzogen. Durch den Wegfall der damit verbundenen Anforderungen werden die Rahmenbedingungen für die Verarbeitung der pseudonymisierten Log-Daten stark vereinfacht. Das Konzept der technischen Zweckbindung und die dafür notwendigen Vertrauens- und Kontrollbeziehungen werden dargestellt.

Das zur praktischen Umsetzung der Konzepte geeignete Software-Paket *Pseudo/CoRe* wird vorgestellt. Es läßt sich unter Wahrung der für die technische Zweckbindung notwendigen Vertrauens- und Kontrollbeziehungen in existierende Unix-Systeme einbetten. Es wird gezeigt, wie *Pseudo/CoRe* Audit-Daten pseudonymisiert und welche Anwendungsmöglichkeiten bestehen. Die Analyse von *Pseudo/CoRes* Laufzeitverhalten zeigt, daß Datenschutz für Log-Daten durch Pseudonymisierung praktikabel ist.

1 Datenschutz-Anforderungen

Nach deutschem Datenschutz-Recht unterliegt die Erhebung, Speicherung und Verarbeitung personenbezogener Daten einem Erlaubnisvorbehalt. Liegt ein gesetzlicher Erlaubnistatbestand oder die Einwilligung des Betroffenen vor, hat der Datenverwender weitreichende Pflichten gegenüber dem Betroffenen, u.a. die Zweckbindung, die Unterrichts-, Lösch- und Meldepflicht [7].

*Die beschriebenen Arbeiten werden derzeit zum Teil von der Deutschen Forschungsgemeinschaft gefördert unter Bi 311/10-2.

Je nach der Art eines Dienstes gelten bei der Verarbeitung personenbezogener Daten ein oder mehrere verschiedene Datenschutzgesetze, die im Detail verschiedene Forderungen und Sanktionen vorsehen [8]. Beispielsweise ist ein Email-Dienst ein Teledienst, während ein WWW-Dienst sowohl ein Tele- als auch ein Mediendienst ist.

Insbesondere die personenbezogenen Anfragen der Nutzer an einen Dienst gehören zu den *Nutzungsdaten*. Ihre Erhebung, Verarbeitung und Nutzung ist “[...] nur soweit dies zur Inanspruchnahme des Teledienstes notwendig ist und nur solange die Nutzung andauert [...]” gestattet. Der Personenbezug ist meist gegeben, z.B. wenn die Anfrage die IP-Adresse des Dienst-Nutzers oder seine Nutzerkonto-Kennung enthält [8, 9].

Insofern ist die von vielen Dienstanbietern angewandte Praxis der Speicherung von Nutzungsdaten in Form von Log- bzw. Audit-Daten problematisch. Im folgenden wird stets der Begriff *Audit-Daten* verwendet. Er bezeichnet die vom Dienst erhobenen Daten über Ereignisse im System, die häufig in ihrer Gesamtheit den Verlauf der Nutzung und des Dienstes dokumentieren, also auch das Verhalten der Nutzer. Audit-Daten werden vielerorts auf Vorrat erhoben und gespeichert. Das Ziel ist die Analyse der Audit-Daten für verschiedene Zwecke. Ein üblicher Zweck ist die Entdeckung von Mißbrauch, welcher dem Urheber zwecks Rechtsverfolgung zugerechnet werden soll, also Intrusion-Detection mit anschließender Incident-Response. Insbesondere außerhalb des Sicherheits-Management-Bereichs sind weitere Zwecke für die Erhebung und Speicherung personenbezogener Audit-Daten denkbar, sind aber hier nicht Gegenstand der Diskussion, z.B. Direktmarketing.

Organisationen stellen ihren Mitarbeitern häufig Dienste für betriebliche bzw. dienstliche Zwecke zur Verfügung. Insbesondere wenn dabei eine private Nutzung geduldet wird, kann durch Audit-Daten das Recht der Mitarbeiter auf informationelle Selbstbestimmung verletzt werden. Zusätzlich hat der Betriebsrat nach deutschem Recht ein Mitbestimmungsrecht bei der Einführung von Technologien, die zur Überwachung der Mitarbeiterleistung geeignet sind [9]. Die Erhebung und Speicherung von Audit-Daten bzw. Nutzungsdaten durch die den Mitarbeitern zur Verfügung gestellten Dienste ermöglicht vielfältige Analysen, auch im Hinblick auf die Arbeitsleistung.

Aufgrund der komplexen rechtlichen Situation und den datenschutz-gesetzlichen Einschränkungen bei der Erhebung, Speicherung und Verarbeitung personenbezogener Daten gestaltet sich der gesetzeskonforme Einsatz von Audit-Daten-gestützten Schutzmaßnahmen wie etwa Intrusion-Detection für viele Dienstanbieter äußerst schwierig.

1.1 Fairer Interessenausgleich

Es besteht ein Spannungsfeld zwischen dem Interesse einzelner Nutzer an Datenschutz und Anonymität einerseits und der Zurechenbarkeit andererseits, um im Mißbrauchsfall die Interessen anderer beteiligter Parteien schützen zu können. Wie etwa die Diskussion in [10, 11, 12] deutlich macht, kann eine für die beteiligten Parteien zufriedenstellende Lösung i.a. nicht darin bestehen, eine der beiden Anforderungen zugunsten der anderen vollständig aufzugeben. Vielmehr scheint ein fairer Ausgleich der Interessen aller beteiligter Parteien unter Berücksichti-

gung der jeweiligen Anwendungssituation erstrebenswert. Dieser Beitrag zeigt, wie ein fairer Interessenausgleich im Hinblick auf die personenbeziehbaren Audit-Daten eines Dienstes gestaltet werden kann.

“§3 Abs. 4 TDDSG und §12 Abs. 5 MDStV: fordern “Gestaltung und Auswahl technischer Einrichtungen für Tele(Medien)Dienste an dem Ziel auszurichten, keine oder so wenig personenbezogene Daten wie möglich zu erheben, zu verarbeiten und zu nutzen.” Diese Anforderung wird durch die in §4 Abs. 1 TDDSG und §13 Abs. 1 MDStV enthaltene Verpflichtung der Diensteanbieter konkretisiert, die Inanspruchnahme von Telediensten und Mediendiensten sowie ihre Bezahlung anonym oder unter Pseudonym zu ermöglichen, soweit dies technisch möglich und zumutbar ist.” [7]. Anonymität und Pseudonymität dienen als Mittel zur Umsetzung von System- und Selbstschutz indem sie Datenvermeidung, Datensparsamkeit und informationelle Selbstbestimmung realisieren [7].

Der in diesem Zusammenhang zentrale Begriff des *Personenbezugs* ist relativ, da die Personenbeziehbarkeit einer Information vom jeweiligen Zusatzwissen abhängt. Dementsprechend gelten die Datenschutzgesetze nur für diejenigen Datenverwender, die durch Zusatzwissen den Bezug der Daten zum Betroffenen herstellen können [7]. Daher kann der oben beschriebene Zielkonflikt zwischen Zurechenbarkeit und Anonymität durch den Einsatz von Pseudonymen fair gelöst werden, indem über die Kontrolle von Zusatzwissen zwischen Regelfall (keine Zurechenbarkeit) und Ausnahmefall (Zurechenbarkeit möglich) unterschieden wird [7].

Mittels Pseudonymen werden personenbezogene Daten so verändert, daß sie ohne Kenntnis der zugehörigen Zuordnungsregel nur mit einem unverhältnismäßig großen Aufwand an Zeit, Kosten und Arbeitskraft einer bestimmten oder bestimmaren natürlichen Person zuordenbar sind, für den Ausnahmefall aber mittels der Zuordnungsregel die Identifizierung der Person ermöglichen [7]. Für Kenner der Zuordnungsregel sind die pseudonymen Daten personenbeziehbar, für diejenigen, die die Zuordnungsregel nicht kennen, sind sie anonym [7].

Da für die Parteien, welche die Zuordnungsregel nicht kennen, die Daten praktisch anonym bzw. nicht personenbezogen sind, fallen die Daten für diese Parteien nicht unter die Datenschutzgesetze [7]. Für diese Parteien entfallen dementsprechend der generelle Erlaubnisvorbehalt, sowie die mit der Verarbeitung verbundenen weitreichenden Pflichten gegenüber den Betroffenen [7] (s. Abschnitt 1).

Im Normalfall sind die Daten für eine Partei nur solange anonym, bis die enthaltenen Pseudonyme ihr gegenüber mit Hilfe der Zuordnungsregel aufgedeckt wurden. Damit die wieder personenbeziehbaren Daten ab diesem Zeitpunkt nicht unter die Datenschutzgesetze fallen, ist der Aufdeckungszweck geeignet einzuschränken und der Aufdeckungszeitpunkt daran auszurichten.

“[...] wenn bereits während der Nutzung vorauszusehen ist, daß gerade diese Daten für die Strafverfolgung erforderlich sind käme die Ausnahme in §6(3) TDDSG zum Zuge [...]” [8], so daß Mißbrauchsverläufe mittels personenbezogener Daten zugerechnet werden können. Daneben führt “[...] das TDDSGÄndG [...] §6 Abs. 8 ein, der es dem Telediensteanbieter erlaubt, personenbezogene Daten desjenigen Nutzers zu speichern, der seinen Teledienst mißbraucht. Die Anhaltspunkte, die zu der Annahme eines solchen Mißbrauchs geführt haben, sind vor

der Speicherung [der personenbezogenen Daten] genau zu dokumentieren. Zu Zwecken der Rechtsverfolgung darf der Diensteanbieter diese personenbezogenen Daten auch über die Speicherfristen hinaus verarbeiten und nutzen. Da das bisherige Gesetz dem Telediensteanbieter generell verbietet, [personenbezogene] Daten zu erheben und zu speichern, wenn nicht einer der wenigen Erlaubnistatbestände vorlag, wäre es dem Telediensteanbieter heute nicht möglich, den Verursacher des Mißbrauchs durch Recherche in seinen Log-Dateien festzustellen.”[8]

Es sollte also erlaubt sein, anonyme Anhaltspunkte zu im voraus definierten Mißbräuchen während deren Verlauf zu speichern, da die anonymen Anhaltspunkte keine personenbezogenen Daten enthalten. In Anlehnung an die obigen Zitate kann man davon ausgehen, daß es erlaubt ist, die Anhaltspunkte eines Mißbrauchsverlaufs mittels Aufdeckung der enthaltenen Pseudonyme zurechenbar zu machen, sobald der anonyme Verlauf so weit voranschreitet, daß ein hinreichender Anfangsverdacht für einen Mißbrauch vorliegt. Wird die beschriebene Aufdeckungsbedingung technisch unumgebar durchgesetzt, sprechen wir von *technischer Zweckbindung* [4]. Die Aussagen zur hier beschriebenen Vorgehensweise stellen allein die Meinung des Autors dar und können nicht eine Rechtsberatung ersetzen.

Die oben beschriebene Aufdeckung wird im folgenden als *geordnete Aufdeckung* bezeichnet, während die Aufdeckung durch das Ausnutzen von Pseudonym-Verkettbarkeit und externem Zusatzwissen als *ungeordnete Aufdeckung* bezeichnet wird. Sowohl für den oben beschriebenen Fall der geordneten als auch für den Fall der ungeordneten Pseudonym-Aufdeckung sind geeignete Vorsorgeregelungen vorzusehen [7]. Dazu gehört die Herstellung von Transparenz gegenüber dem Nutzer, z.B. darüber, daß seine Anonymität beim Mißbrauch aufgehoben wird. Desweiteren sind Maßnahmen zur Sicherung der Pseudonymitätseigenschaft vorzusehen. In diesem Kontext sind die Ausführungen in Abschnitt 2 zum Vertrauensmodell zu sehen.

2 Vertrauens- und Kontrollbeziehungen

In [6] wurden Architekturen vorgestellt, analysiert und verglichen, die Nutzer-Anonymität gegenüber den *Sicherheits-Administratoren* eines Dienstes herstellen, welche Beobachtungen ausschließlich auf der Basis der vom Dienst gelieferten Audit-Daten machen können. Eine zentrale Anforderung bei der Analyse von Audit-Daten hinsichtlich Anhaltspunkten für Mißbrauch ist die zeitnahe Pseudonym-Aufdeckbarkeit zwecks Zurechenbarkeit. Dies ist mittels technischer Zweckbindung der geordneten Pseudonym-Aufdeckung erreichbar. Unter dem Gesichtspunkt der Praktikabilität sind die Unabhängigkeit der Lösung vom Nutzer und die Unabhängigkeit von aufwendigen Infrastrukturen entscheidend. Damit die Verarbeitung der Audit-Daten nicht den Datenschutzgesetzen unterliegt, besteht dienstseitig ein Interesse daran, daß die Audit-Daten bei den Sicherheits-Administratoren stets anonym vorliegen, also unabhängig von Maßnahmen, die der Nutzer im Hinblick auf Anonymität ergreifen kann. Der zum Aufbau einer für Anonymität benötigten Infrastruktur erforderliche Aufwand und deren Rahmenbedingungen haben einen beträchtlichen Einfluß darauf, in welchem Zeitrahmen die Infrastruktur auf breiter Basis zur Verfügung gestellt werden kann. Es ist dienstseitig also von Vorteil, wenn die Anonymisierung der Audit-Daten unabhängig von Nutzermaßnahmen und

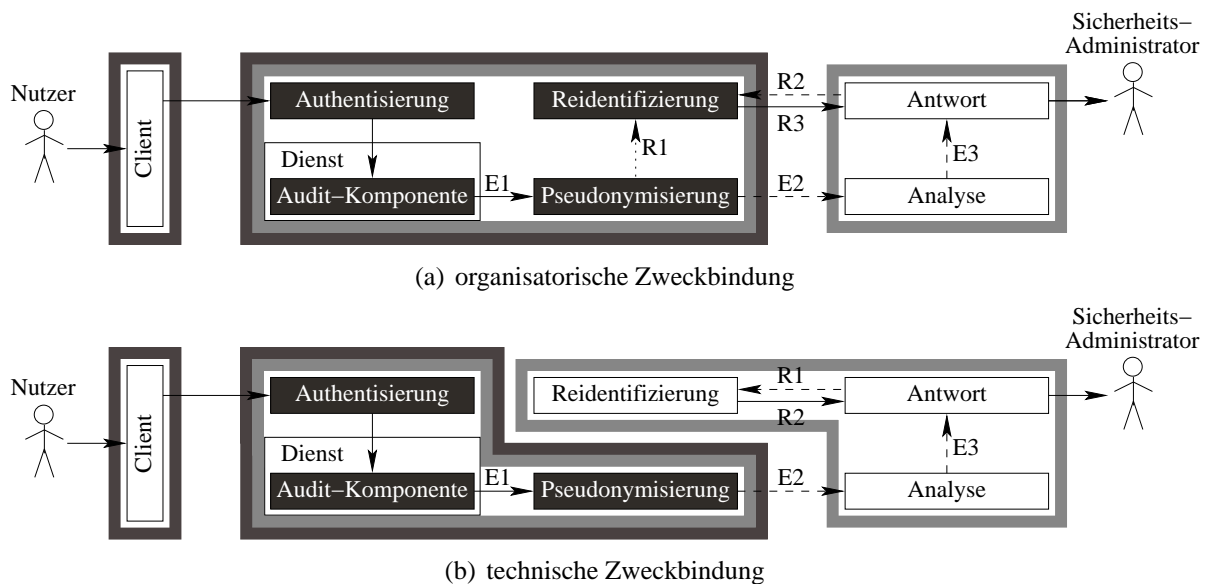


Abbildung 1: Zweckbindung der geordneten Aufdeckbarkeit

Infrastrukturen implementierbar ist. Ein Vergleich verschiedener Architekturen zeigt, daß diese Anforderungen gemeinsam nur erfüllbar sind, wenn die Audit-Daten nach ihrer Erhebung pseudonymisiert werden [6]. Daher werden im folgenden nur ebensolche Systeme betrachtet.

Die Audit-Daten werden normalerweise von der *Audit-Komponente* des Dienstes erhoben und der *Audit-Analyse* der Sicherheits-Administratoren verfügbar gemacht. Diese erzeugt entsprechend des *Analyse-Zwecks Einzelberichte* und sendet sie an die *Antwort-Einheit*, welche wiederum geeignet auf Einzelberichte reagiert, z.B. indem sie den Sicherheits-Administrator unterrichtet. Ein Einzelbericht kann einen *Analyse-Kontext* enthalten, der eine Untermenge der Audit-Daten ist. Eine konkrete Instanz dieses Szenarios wäre ein Intrusion-Detection-System, dessen Analyse-Zweck das Entdecken bekannter und durch die Dienstanwender verursachter Anfangsverdachte für *Schutzzielverletzungen* bzw. Mißbräuche ist. Ein Einzelbericht ist ein *Alarm*, der als Analyse-Kontext etwa die *Anhaltspunkte* für den Anfangsverdacht in der Form von *Audit-Datensätzen* enthält, die den Verlauf der Schutzzielverletzung dokumentieren (s. Abschnitt 1.1). Intrusion-Detection-Systeme befinden sich von Grund auf in dem in Abschnitt 1.1 beschriebenen Interessenkonflikt (s. auch [10]). Abb. 1 zeigt die Architekturen von Systemen, die personenbezogene Audit-Daten nach ihrer Erhebung anonymisieren.

In Abb. 1 zeigen die *soliden Pfeile* die Flußrichtung personenbezogener Merkmale an. Die *gestrichelten Pfeile* zeigen die Flußrichtung der durch Pseudonymisierung anonymen Merkmale an. Schließlich zeigen die *gepunkteten Pfeile* die Flußrichtung der für die Pseudonymisierung notwendigen Information an, die als *Zuordnungsregel* bezeichnet wird. Jede fette graue Umrahmung schließt einen Bereich ein, in dem die Interessen einer Partei durchgesetzt werden. In einem solchen Bereich dürfen jene Parteien keine Kontrolle ausüben, deren Interessen mit den im Bereich durchgesetzten Interessen im Konflikt stehen. Dabei stehen die dunkelgrauen Umrahmungen für das Nutzerinteresse Anonymität und die hellgrauen Umrahmungen für das Interesse der Sicherheits-Administratoren an Zurechenbarkeit. Dunkel ausge-

füllte Kästen setzen gemeinsam den fairen Interessenausgleich durch. Sie befinden sich gerade in den doppelt umrahmten Bereichen, also dort, wo konfligierende Interessen durchgesetzt werden. Dementsprechend dürfen die ausgefüllten Kästen nicht vom Nutzer und auch nicht vom Sicherheits-Administrator kontrolliert werden, sondern müssen von einer Partei kontrolliert werden, der beide vertrauen können. Eine naheliegende Wahl für diese vertrauenswürdige Partei ist der Datenschutz-Beauftragte der Organisation, die den Dienst betreibt. Bestehen allerdings Zweifel daran, daß der Datenschutz-Beauftragte beide Interessen zu einem fairen Ausgleich bringt, muß eine andere, ggf. externe Partei gewählt werden.

Abb. 1a zeigt im Vergleich mit Abb. 1b, wie durch technische Zweckbindung die notwendigen Kontrollverhältnisse vereinfacht werden. Bei der technischen Zweckbindung der Aufdeckbarkeit wird den Pseudonymen die zweckgebunden geschützte Zuordnungsregel beigelegt (s. E2 in Abb. 1b), so daß diese nicht mehr direkt dem Reidentifizierer übermittelt werden muß (vgl. R1 in Abb. 1b). Die Reidentifizierung ist so unumgebar nur noch entsprechend dem Zweck der geordneten Aufdeckung möglich. Demgemäß muß der Nutzer der Partei, welche die Reidentifizierung kontrolliert, nicht mehr vertrauen. Da also die Sicherheits-Administratoren nicht mehr von der Kontrolle des Reidentifizierers ausgeschlossen sind, können sie die Reidentifizierung selbst kontrollieren und die Reidentifizierung unverzüglich durchführen, sobald der entsprechende Zweck vorliegt. Erfordert der Zweck der Audit-Daten-Analyse eine rasche geordnete Aufdeckbarkeit, läßt sich dies nur mittels technischer Zweckbindung erreichen. Die organisatorische Zweckbindung der geordneten Aufdeckbarkeit erfordert die Kooperation des Datenschutz-Beauftragten. In der Regel findet die Zweckprüfung manuell statt, so daß sich die Aufdeckung verzögert, wenn die verantwortliche Person nicht verfügbar ist.

Eine weitere Anforderung betrifft die Performanz der Pseudonym-Erzeugung. Je nach Sorte des Dienstes, der die Audit-Daten erzeugt, kann ein extrem hohes Aufkommen zu bewältigen sein, insbesondere beim Dienst Betriebssystem, wenn es für Intrusion-Detection Systemrufe als Audit-Datensätze speichert. Die Pseudonym-Erzeugung findet idealerweise on-the-fly statt und sollte daher einen dem Datenaufkommen angemessenen Durchsatz erreichen. Im Idealfall findet die Anonymisierung von Audit-Daten dort statt, wo die Audit-Daten erhoben werden, nämlich auf dem Gerät, das die Nutzeranfragen zur Diensterbringung verarbeitet. Damit der Dienst nicht ausgebremst wird ist es wichtig, daß die Pseudonym-Erzeugung nicht den überwiegenden Teil der Prozessor-Ressourcen bindet. Aufwendige kryptographische Verfahren für die Pseudonym-Erzeugung entfallen daher für die on-the-fly-Pseudonymisierung.

2.1 Verwandte Ansätze

Folgende Ansätze wurden verglichen [6]: *Anonymouse Log File Anonymizer* [13], *Jaeger-Anonymisierer* [8], *Lundin Firewall Audit Anonymisierer* [14], *WebWasher* [15], *Intrusion Detection and Avoidance (IDA)* [16], *Adaptive Intrusion Detection (AID)* [9, 17], sowie *Pseudonymization with Conditional Reidentification (Pseudo/CoRe)*. Einen detaillierteren Überblick zu den Systemen *Lundin Firewall Audit Anonymizer*, *IDA* und *AID* bieten [1, 2]. Es stellt sich heraus, daß die Ansätze entweder keine Zurechenbarkeit durch geordnete Pseudonym-Aufdeckung unterstützen, oder die notwendigen Vertrauens- und Kontrollbeziehungen wur-

den beim Entwurf nicht vollständig berücksichtigt, so daß der Sicherheits-Administrator unter Umgehung der Zweckbindung direkten Zugriff auf die Zuordnungsregel erlangen kann. Bei einigen Ansätzen wurde ein ungeeignetes Verfahren zur Implementierung des 4-Augen-Prinzips gewählt, so daß der aufgeteilte Dechiffrier-Schlüssel nach der ersten Aufdeckung zumindest einer der beiden Entitäten bekannt ist [6].

Der Vergleich [6] ergibt, daß nur der *Pseudo/CoRe*-Ansatz in der Lage ist, Audit-Daten im Sinne eines fairen Interessenausgleichs durch technische Zweckbindung zu anonymisieren. Dazu tragen die klar definierten und umsetzbaren notwendigen Vertrauens- und Kontrollverhältnisse beim Einsatz von *Pseudo/CoRe* bei (s. Abschnitt 3.1). *Pseudo/CoRe* ist auch der einzige Ansatz, der die technische Zweckbindung konsequent durchsetzt. Dies gilt nicht nur für die geordnete Pseudonym-Aufdeckung, sondern auch für die automatischen Pseudonym-Wechsel. Erstens definiert der Aufdeckungszweck verschiedene Mißbrauchs-Szenarien. In jedem besitzt ein Nutzer ein anderes Pseudonym [3]. Zweitens werden Pseudonyme automatisch unaufdeckbar, wenn die mit ihnen verbundenen Verdachtsmomente sich nicht bestätigen. Optional wird ebenfalls eine organisatorische Zweckbindung bei der geordneten Pseudonym-Aufdeckung unterstützt.

3 *Pseudo/CoRe*

Die Implementierung von *Pseudo/CoRe* ist für Unix-Systeme ausgelegt und wurde erfolgreich unter folgenden Betriebssystemen getestet: Solaris, OpenBSD und Linux. Der Pseudonymisierer kann benutzt werden, um Audit-Daten im ASCII-Format zu pseudonymisieren, z.B. *Syslog*-Audit-Daten und Web-Server-Audit-Daten. Insbesondere die Integration mit *Syslog* wird speziell unterstützt, da *Syslog* auf allen wichtigen modernen Unix-Systemen verfügbar ist. Auch Windows-Systeme und sehr viele Netzwerk-Komponenten können in eine *Syslog*-Infrastruktur eingebunden werden. Dementsprechend hoch ist die erreichbare Abdeckung anfallender Audit-Daten durch *Pseudo/CoRe* [2].

Pseudo/CoRe besteht aus mehreren Komponenten, die entweder der Pseudonymisierung von Audit-Daten dienen (`pseudonymizer` und `shared`) oder der Reidentifizierung von Audit-Daten dienen (`reidentifizier` und `combined`). Weitere Werkzeuge dienen der Einbettung des `pseudonymizers` in den Audit-Datenstrom (`wrapper`, `redirector` und `rlogger`).

3.1 Vertrauens- und Kontrollbeziehungen bei *Pseudo/CoRe*

Der *Pseudo/CoRe*-Ansatz bietet technische Zweckbindung bei der geordneten Pseudonym-Aufdeckung (vgl. Abschnitt 2). Dementsprechend liegt das Vertrauen der Nutzer bzw. Sicherheits-Administratoren hinsichtlich eines fairen Ausgleichs ihrer Interessen Anonymität bzw. Zurechenbarkeit beim Datenschutz-Beauftragten. In diesem Sinne kontrolliert der Datenschutz-Beauftragte die Audit-Komponenten und Pseudonymisierung, während der Sicherheits-Administrator die Reidentifizierung kontrolliert (vgl. Abb. 1b und Abb.3). Der

Datenschutz-Beauftragte modelliert im Sinne der Nutzerinteressen und möglicherweise in Zusammenarbeit mit dem Betriebsrat im voraus das Wissen des Pseudonymisierers hinsichtlich zu pseudonymisierender personenbezogener Merkmale. In Zusammenarbeit mit den Sicherheits-Administratoren modelliert der Datenschutz-Beauftragte a priori die Anfangsverdachte, die eine Pseudonym-Aufdeckung rechtfertigen. Dieses Wissen stellt der Datenschutz-Beauftragte den von ihm kontrollierten Audit-Komponenten und den von ihm kontrollierten Pseudonymisierern in Form von Konfigurationsdaten zur Verfügung.

Diese Vorgehensweise ist geeignet, um die widerstreitenden Interessen im Vorfeld detailliert zu erfassen und dann zu implementieren, wenn eine Einigung erzielt wurde. Die Durchsetzung der so im vorhinein genau spezifizierten Interessen soll im laufenden Betrieb automatisch und sicher durch den Pseudonymisierer erfolgen, sofern die oben spezifizierten Kontrollverhältnisse wirksam sind. Entsprechend der technischen Zweckbindung sind die Pseudonyme ausschließlich dann geordnet aufdeckbar, wenn sie zu den Anhaltspunkten eines erfüllten vorher definierten Anfangsverdachts gehören. So ist die Zurechenbarkeits-Anforderung der Sicherheits-Administratoren erfüllt, wenn ein Anfangsverdacht erfüllt ist. Im Normalfall, also ohne erfüllten Anfangsverdacht, sind die Pseudonyme nicht geordnet aufdeckbar, wodurch die Anonymitäts-Anforderung der Nutzer gewahrt bleibt.

Generell sind Audit-Daten integer, solange kein Angreifer hinreichende Privilegien auf der erhebenden Maschine erlangt hat, welche es ihm erlauben, Audit-Daten zu löschen oder zu korrumpieren. Entsprechendes gilt für die auf derselben Maschine erzeugten Pseudonyme. Audit-Datensätze sind daher unmittelbar nach ihrer Erzeugung zu pseudonymisieren und zwecks Analyse aus dem Zugriffsbereich des Angreifers abzutransportieren. Demgemäß muß dieser Vorgang on-the-fly durchführbar sein und darf nicht zu einem Flaschenhals werden (s. Abschnitt 5). Unter der Annahme, daß Anfangsverdachte entdeckt werden, bevor der Angreifer hinreichende Privilegien erlangen kann, sind bei einem Angriff die Pseudonyme integer und aufdeckbar.

Die Audit-Daten sollten lokal auf der erzeugenden Maschine pseudonymisiert werden. Ist dies nicht möglich, muß der Datenschutz-Beauftragte einen sicheren Kanal zwischen der Audit-Komponente und dem `pseudonymizer` etablieren.

3.2 Alternativen bei der Einbettung von *Pseudo/CoRe*

Abb. 2 bildet abstrakt die möglichen Audit-Komponenten eines Solaris-Systems ab. Audit-Datensätze können auf ihrem Pfad von der Audit-Komponente bis zur Daten-Sinke, hier Audit-Dateien, anonymisiert werden. Die entsprechenden möglichen Platzierungen für Pseudonymisierer sind in Abb. 2 dargestellt, wobei die bisher mittels *Pseudo/CoRe* implementierbaren Platzierungen dunkel markiert sind.

Die einfachste Möglichkeit zur Einbettung des `pseudonymizers` besteht darin, eine Audit-Komponente in eine Pipe schreiben zu lassen, die der `pseudonymizer` liest (s. *P* nahe den Audit-Dateien *Host-Audit*, *App.-Audit* und *Net-Audit* in Abb. 2). Diese Vorgehensweise bietet

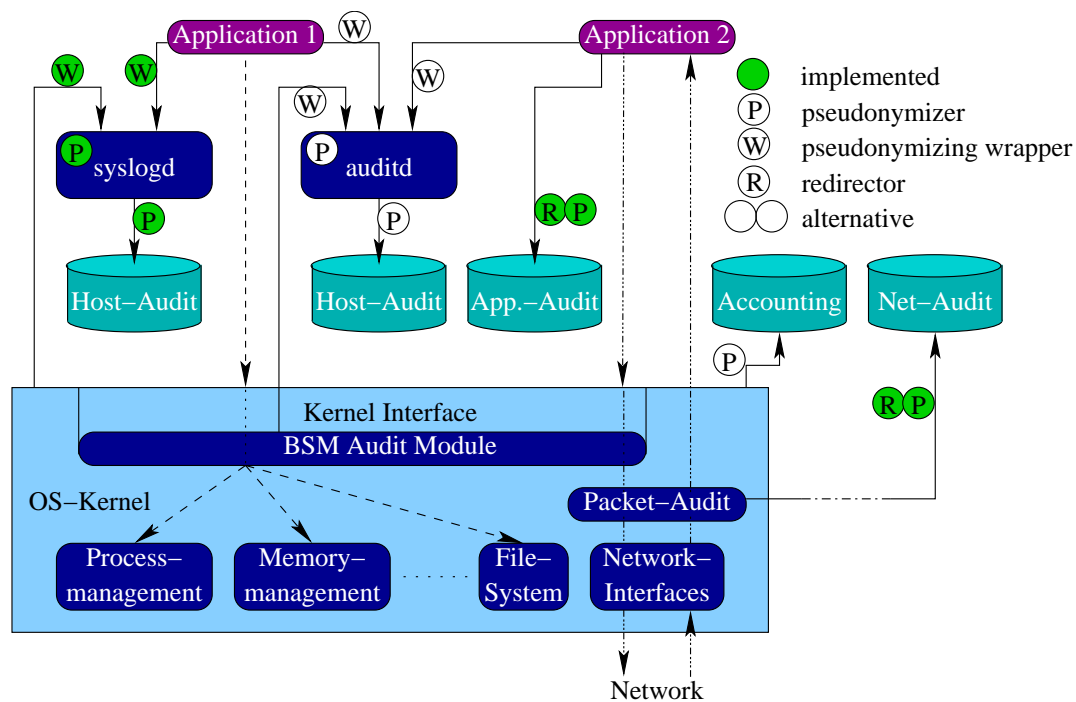
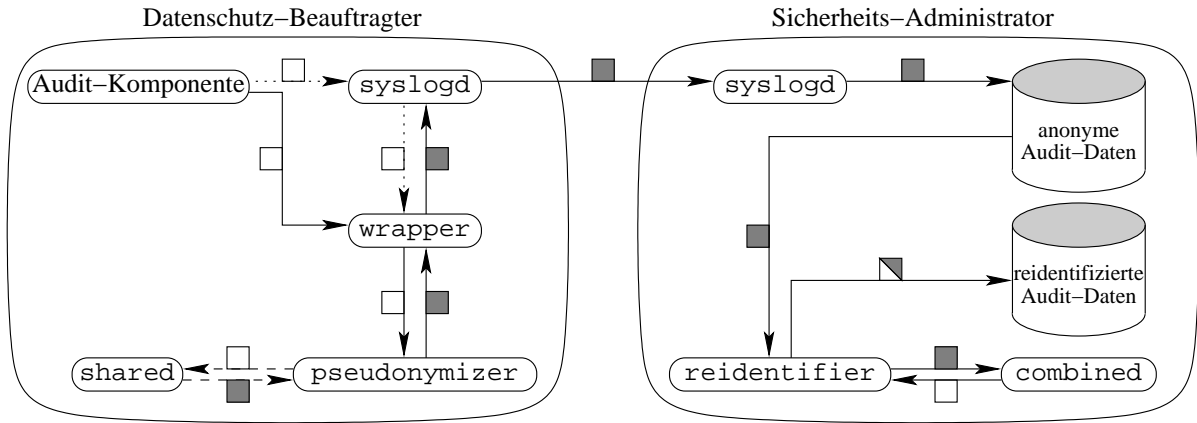


Abbildung 2: Pseudonymisierung von Solaris-Audit-Daten

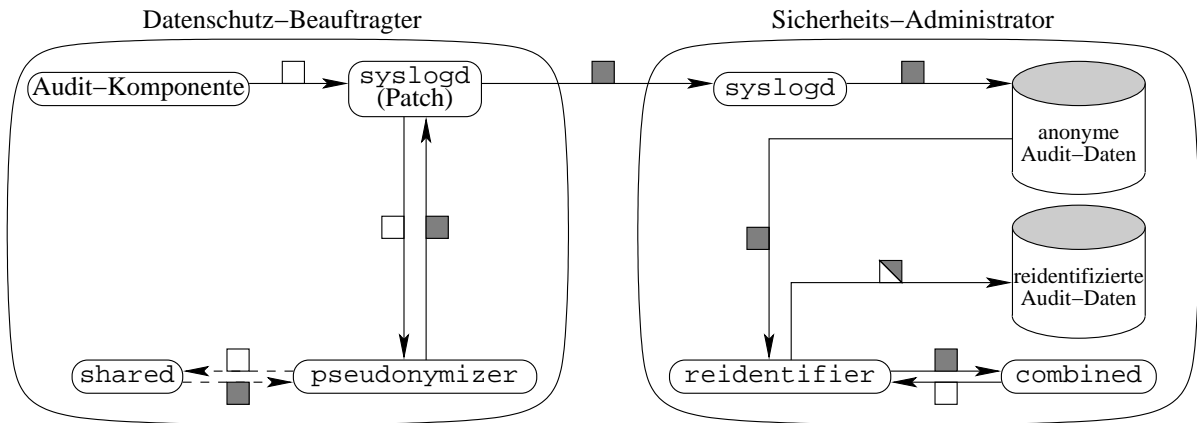
sich bei Applikationen an, die ihre Audit-Daten direkt in eine Datei schreiben, z.B. Web-Server (s. *P* nahe der Audit-Datei *App.-Audit* in Abb. 2). Selbstverständlich können auch bereits archivierte Audit-Daten nachträglich pseudonymisiert werden. Sollen alle Audit-Daten über *Syslog* konsolidiert werden, läßt man den *redirector* aus der Pipe lesen, in welche die Audit-Komponente schreibt (s. *R* nahe den Audit-Dateien *App.-Audit* und *Net-Audit* in Abb. 2). Der *redirector* deponiert die Audit-Daten mittels des *Syslog*-API.

Für die Integration von *Pseudo/CoRe* mit *Syslog* werden drei Möglichkeiten unterstützt. Erstens ist es möglich, den *syslogd* in eine Pipe schreiben zu lassen, die der *pseudonymizer* liest (s. *P* nahe der Audit-Datei *Host-Audit* in Abb. 2). Die pseudonymisierten Audit-Daten können anschließend mittels *rlogger* an einen *syslogd* auf einem anderen Rechner geleitet werden (s. Abb. 3c, der *rlogger* ist aus Platzgründen nicht explizit dargestellt). Da der *pseudonymizer* bei dieser Architektur nicht die Priorität der *Syslog*-Datensätze ermitteln kann, wird vom *rlogger* bei der Weiterleitung eine konfigurierbare Priorität festgelegt. Um die Audit-Datensätze vom empfangenden *syslogd* nach Prioritäten zu differenzieren, würden entsprechend viele Instanzen von Pipes, *pseudonymizer* und *rlogger* benötigt. Ein weiterer Nachteil dieser Einbettung ist, daß der *syslogd* ggf. identische aufeinander folgende Audit-Datensätze zusammenfaßt und der *pseudonymizer* darauf nicht adäquat reagieren kann. Es ist daher sinnvoll, den *pseudonymizer* so im Audit-Datenstrom einzubinden, daß er Zugriff auf die Priorität jedes Audit-Datensatzes hat.

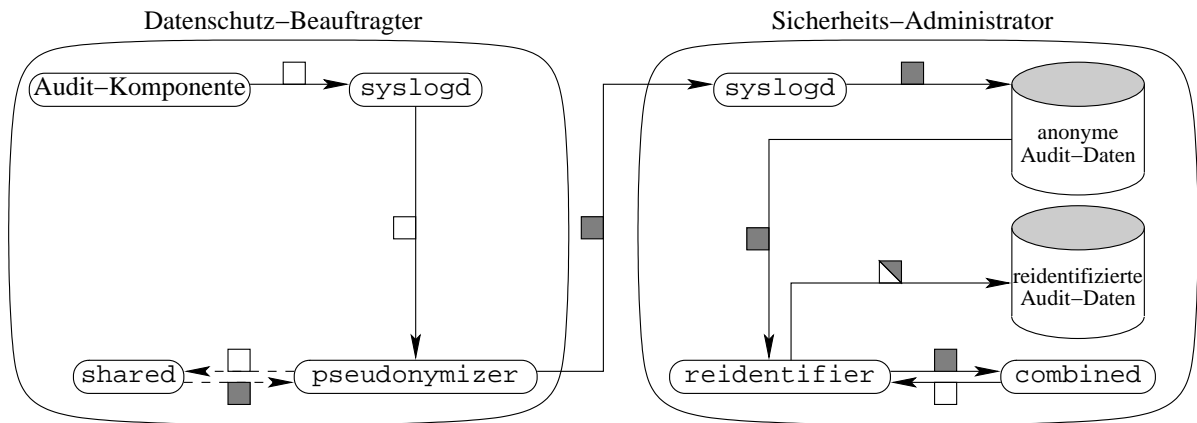
Zweitens kann der *pseudonymizer* direkt vom *syslogd* genutzt werden, wie in Abb. 3b dargestellt (vgl. *P* im *syslogd* in Abb. 2). Der vorhandene *syslogd* kann mittels eines Patches entsprechend angepaßt werden. Liegt der Source-Code nicht vor, kann *syslogd* durch ei-



(a) Wrapper



(b) Patch



- personenbezogene Merkmale
- pseudonymisierte Merkmale
- ▣ personenbezogene und pseudonymisierte Merkmale
- ungeschützte Kommunikation
- ⋯→ ungeschützte, nicht umgeleitete Kommunikation
- - -> SSL-geschützte Kommunikation

(c) Pipes

Abbildung 3: Pseudo/CoRe-Architekturen für die Integration mit Syslog

ne angepaßte Version ersetzt werden. Nachteile dieses Ansatzes sind die Verfügbarkeit des Source-Codes bzw. einer angepaßten Version, welche dieselben Features hat, wie der vorhandene `syslogd` (z.B. bei Solaris).

Drittens und letztens kann der `wrapper` genutzt werden, um Audit-Datensätze dem `pseudonymizer` zuzuführen, bevor sie den `syslogd` erreichen (s. *W* oberhalb vom `syslogd` in Abb. 2). Lokal erzeugte Audit-Daten liest der `wrapper` von `/dev/log`. Audit-Daten von den `syslogds` anderer Rechner werden lokal per UDP-Port-Forwarding von UDP-Port 514 so umgeleitet, daß sie vom `wrapper` empfangen werden. Audit-Daten, die der `wrapper` nicht abfangen kann, werden ihm vom entsprechend konfigurierten `syslogd` über eine Pipe zu Verfügung gestellt (s. gestrichelte Pfeile in Abb. 3a), z.B. OpenBSD-Kernel-Audit-Daten, die der OpenBSD-`syslogd` immer von `/dev/klog` liest. Der `wrapper` läßt die personenbeziehbaren Audit-Daten vom `pseudonymizer` pseudonymisieren und leitet die anonymen Audit-Daten an den `syslogd` weiter.

Die drei *Pseudo/CoRe*-Architekturen für *Syslog* in Abb. 3 sind so entworfen, daß sie die notwendigen Kontrollbeziehungen ermöglichen (vgl. Abschnitt 3.1) und personenbeziehbare Audit-Daten lokal vom `pseudonymizer` anonymisiert werden, bevor sie in einer Audit-Datei gespeichert werden. Die anonymen Audit-Daten werden zwecks Analyse aus dem Kontrollbereich des Datenschutz-Beauftragten heraus an eine zentrale Sammelstelle weitergeleitet, die sich im Kontrollbereich der Sicherheits-Administratoren befindet. Dort können die anonymen Audit-Daten außerhalb des Geltungsbereichs der Datenschutzgesetze analysiert werden. Dafür können die bisher verwendeten (Legacy-)Tools zum Einsatz kommen, da sich der `pseudonymizer` so konfigurieren läßt, daß er das Format und die Verkettbarkeit der Audit-Daten beibehält (s. Abschnitt 4). Enthält der von der Analyse gelieferte Alarm die Anhaltspunkte bzw. Audit-Datensätze auf denen er basiert, können diese mittels des `reidentifiers` aufgedeckt werden. Voraussetzung dafür ist, daß der Datenschutz-Beauftragte Aufdeckbarkeit vorgesehen hat und die entsprechenden Kontexte für diese Anhaltspunkte definiert hat. Der `reidentifier` könnte z.B. bei einem Alarm automatisch aufgerufen werden.

3.3 Funktionsweise von *Pseudo/CoRe*

Ein gemäß Abschnitt 3.2 in den Audit-Datenstrom eingebetteter `pseudonymizer` erhält sequentiell Audit-Datensätze. Jeden empfangenen Audit-Datensatz untersucht er daraufhin auf Merkmale, die er gemäß Konfiguration anonymisieren soll. Das hierfür notwendige Parsing verwendet reguläre Suchausdrücke. Wurde ein zu anonymisierendes Merkmal gefunden, ersetzt der `pseudonymizer` es durch ein Pseudonym. Gemäß Konfiguration wird dabei das Format und die Verkettbarkeit des Pseudonyms berücksichtigt. Außerdem entscheidet der `pseudonymizer` gemäß Konfiguration, ob das Pseudonym des Merkmals aufdeckbar sein soll. Ist dies der Fall, bestimmt der `pseudonymizer` die Anfangsverdachts-Kontexte, in denen das Pseudonym aufdeckbar sein soll und kontaktiert den `shared` mit der Anfrage, entsprechendes kryptographisches Material zu liefern.

Ein hinreichender Anfangsverdacht ist dabei definiert als ein Schwellenwert über gewichteten Anhaltspunkten für potentiell angriffsbezogene Aktivitäten. Überschreiten die Anhalts-

punkte den Schwellenwert, so sollen die enthaltenen Pseudonyme aufdeckbar sein, sonst aber nicht. Grundlage für die technische Realisierung ist in unserem Ansatz das Shamir'sche Schwellenwert-Schema zur informationstheoretisch sicheren Geheimnisteilung [18]. Ein in einen Anhaltspunkt eingebettetes und zu pseudonymisierendes personenbeziehbares Merkmal wird zunächst verschlüsselt, um es zu verbergen. Dann wird das Shamir'sche Schema leicht modifiziert eingesetzt, um den kryptographischen Dechiffrier-Schlüssel des verschlüsselten personenbeziehbaren Anhaltspunkts-Merkmals kryptographisch in Anteile aufzuteilen. Jedes Vorkommen des verschlüsselten personenbeziehbaren Merkmals, das einem bestimmten Anfangsverdacht-Kontext zugeordnet ist, erhält einen eigenen Anteil des Dechiffrier-Schlüssels. Die Verkettbarkeit der Schlüssel-Anteile besteht über ihr verschlüsseltes personenbeziehbares Merkmal. Überschreitet die Anzahl der vorliegenden Anteile den Schwellenwert des Anfangsverdachts, so kann der Dechiffrier-Schlüssel per Lagrange-Interpolation zurückgewonnen und das personenbezogene Merkmal entschlüsselt werden [3].

Der `shared` kapselt die kryptographischen Primitiven, die für die geordnete Aufdeckbarkeit der Pseudonyme notwendig sind. Dabei kommt für die symmetrische Chiffrierung mit Blowfish und für das Hashen mit SHA1 die OpenSSL Crypto Library [19] zum Einsatz. Die Geheimnisteilung ist mit Hilfe der GNU Multiple Precision Arithmetic library (GMP) [20] und der Number Theory Library (NTL) implementiert [21]. Da personenbeziehbare Daten zwischen `pseudonymizer` und `shared` ausgetauscht werden, wird die Kommunikation mittels der OpenSSL SSL/TLS library [22] geschützt, wenn der `shared` nicht lokal angesprochen wird. Das vom `shared` gelieferte kryptographische Material fügt der `pseudonymizer` als separate und entsprechend gekennzeichnete Datensätze den Audit-Daten hinzu. Soll die geordnete Aufdeckbarkeit anonymer Audit-Daten nachträglich vollständig unterbunden werden, können die Datensätze mit dem kryptographischen Material durch Filtern entfernt werden.

Der `reidentifizier` dient der Reidentifizierung der anonymen Audit-Datensätze, die er als Eingabe erhält. Er deckt alle Pseudonyme auf, die einen vor der Pseudonymisierung definierten hinreichenden Anfangsverdacht erfüllen. Er kann also entweder eine komplette Audit-Datei soweit möglich aufdecken, oder auch nur die mit einem Alarm gelieferten Datensätze bzw. Anhaltspunkte. Dafür lokalisiert der `reidentifizier` das zu jedem aufzudeckenden Pseudonym gehörige kryptographische Material und ordnet es gemäß Konfiguration den entsprechenden Anfangsverdachts-Kontexten zu. Ist ein Anfangsverdacht eines Pseudonyms erfüllt, d.h. der zugehörige im voraus definierte Schwellenwert ist überschritten, kann das Pseudonym aufgedeckt werden. Zu diesem Zweck kontaktiert der `reidentifizier` den `combined` und übermittelt ihm das in diesem Anfangsverdachts-Kontext zum Pseudonym gehörige kryptographische Material.

Der `combined` kapselt die kryptographischen Primitiven, die für die geordnete Aufdeckung der Pseudonyme notwendig sind. Für die symmetrische Dechiffrierung mit Blowfish und das Hashen mit SHA1 wird die OpenSSL Crypto Library [19] verwendet und die Lagrange-Interpolation ist mit Hilfe der GMP [20] implementiert. Personenbeziehbare Daten werden zwischen `reidentifizier` und `combined` ungeschützt ausgetauscht, da der Sicherheits-Administrator sie gemäß der vereinbarten Anfangsverdachte kennen darf.

4 Anwendung von *Pseudo/CoRe*

Nachdem *Pseudo/CoRe* gemäß Abschnitt 3.2 und unter Wahrung der in Abschnitt 3.1 definierten Kontrollbeziehungen in das System eingebettet wurde, kann dem `pseudonymizer` und dem `reidentifizier` das Wissen über zu pseudonymisierende Merkmale und deren Aufdeckbarkeit als Konfiguration zur Verfügung gestellt werden.

Damit der `pseudonymizer` personenbeziehbare Merkmale durch Pseudonyme ersetzen kann, müssen die zu pseudonymisierenden Merkmale spezifiziert werden. Jede Audit-Komponente kann für jeden zu dokumentierenden Ereignis-Typ ein eigenes Format verwenden. Darum werden zunächst für jede Audit-Komponente (s. `FACILITY` in Abb. 4) die verschiedenen von ihr erzeugten Ereignis-Typen mittels regulärer Suchausdrücke definiert (s. `EVENT` in Abb. 4). Jeder Ereignis-Typ kann mehrere zu pseudonymisierende Merkmale enthalten, wobei jedes Merkmal von einem linken und einem rechten Merkmals-Kontext umgeben ist. Die Merkmals-Kontexte werden ebenfalls mittels regulärer Suchausdrücke spezifiziert (s. `LEFT` und `RIGHT` in Abb. 4).

Zur Laufzeit identifiziert der `pseudonymizer` für einen gegebenen Audit-Datensatz zunächst die erzeugende Audit-Komponente sowie den Ereignistyp und erhält so die im Datensatz zu suchenden Merkmals-Kontexte. Nachdem der `pseudonymizer` mittels eines Merkmals-Kontext-Paares ein zu pseudonymisierendes Merkmal lokalisiert hat, kann er es durch ein Pseudonym ersetzen. Die analoge Vorgehensweise wendet der `reidentifizier` an, um die aufzudeckenden Pseudonyme zu lokalisieren.

Für jedes zu ersetzende Merkmal kann das Pseudonym-Format festgelegt werden. Der Typ eines zu ersetzenden Merkmals bestimmt die Syntax des zugehörigen Pseudonyms (s. `TYPE` in Abb. 4). Unterschieden wird zwischen positiven ganzen Zahlen (`INT`), IP-Adressen (`IP`), (mehrstufigen) DNS-Namen (`DNS`) und einfachen Zeichenketten (s. `STRING` in Abb. 4). Für Adresstypen ist einstellbar, wieviele Stufen der Adresse pseudonymisiert werden sollen (`IP BITS`, `DNS LEVELS`). Werden Adressen nur partiell pseudonymisiert, können weiterhin grobe Informationen über ihren Ursprung gewonnen werden. Die Pseudonyme für ganze Zahlen oder Zeichenketten können entweder die Länge des ersetzten Merkmals beibehalten (s. `KEEPLLEN` in Abb. 4) oder eine spezifizierte Länge erhalten (s. `LEN` in Abb. 4), um die Pseudonym-Verkettbarkeit aufgrund der Pseudonym-Länge zu unterbinden.

Des weiteren können die Pseudonyme verkettbar oder unverkettbar erzeugt werden (s. `LINK` und `UNLINK` in Abb. 4). Diese Verkettbarkeit bezieht sich lediglich auf die Pseudonyme, welche die personenbeziehbaren Merkmale in den Audit-Datensätzen ersetzen. Verkettbarkeit ist notwendig, wenn die Analyse der pseudonymisierten Audit-Datensätze es erfordert. Die faktische Verkettbarkeit dieser Pseudonyme hängt jedoch zusätzlich vom ggf. dazugehörigen kryptographischen Material ab (s. Abschnitt 4.1).

Schließlich kann spezifiziert werden, ob die Pseudonyme die Möglichkeit zur geordneten Aufdeckung besitzen (s. `RECOVER` und `NORECOVER` in Abb. 4) und welche Sorten(n) der Zweckbindung dabei gelten. Bei normalem Einsatz sind Pseudonyme ausschließlich unter technischer Zweckbindung aufdeckbar, also wenn sie im Kontext eines erfüllten Anfangsverdachts auftreten. Ein Anfangsverdacht wird definiert durch seinen Schwellenwert (s. `THRESHOLD` in Abb. 4)

sowie die im selben Kontext auftretenden Anhaltspunkt-Merkmale (s. `CONTEXT` in Abb. 4). Mithin kann ein Merkmal verschiedenen Anfangsverdachts-Kontexten zugeordnet werden.

Der Einfluß, den das Auftreten eines gegebenen Anhaltspunkt-Merkmals in einem gegebenen Kontext hat, wird weiter danach spezifiziert, ob mehrmaliges Auftreten signifikant ist (s. `GROW` in Abb. 4), und mit welchem Gewicht das Auftreten den Anfangsverdacht verstärkt (s. `ADD` in Abb. 4). Ist jedes Auftreten eines spezifischen Anhaltspunkt-Merkmals in dem Anfangsverdachts-Kontext signifikant, sollte der Anfangsverdacht bei jedem Auftreten verstärkt werden (`GROW`). Ist lediglich relevant, ob das Merkmal mindestens einmal auftritt, sollte der Anfangsverdacht nur beim ersten Auftreten verstärkt werden (`ONCE`). Durch eine verschiedene Gewichtung kann der Einfluß verschiedener Anhaltspunkte eines Anfangsverdachts-Kontextes differenziert werden. Aktivitäten, die einen aufgetretenen Anhaltspunkt im Sinne eines Anfangsverdachts entkräften, werden entsprechend negativ gewichtet (`DEL`) (s. Abschnitt 4.1).

Stellt sich im nachhinein heraus, daß die anonymen Audit-Daten Anhaltspunkte für einen nicht im voraus spezifizierten Anfangsverdacht enthalten, sind die Pseudonyme aufgrund der technischen Zweckbindung dennoch nicht aufdeckbar. Da die Audit-Daten bereits den `pseudonymizer` und damit den Kontrollbereich des Datenschutz-Beauftragten verlassen haben, ist es nicht praktikabel, nach dem Hinzufügen des neuen Anfangsverdachts zur Konfiguration, die Pseudonyme entsprechend anzupassen. Einen Ausweg bietet in dieser Situation die von *Pseudo/CoRe* optional zusätzlich unterstützte organisatorische Zweckbindung (`ESCROW`). Die Anhaltspunkt-Merkmale können dann nur vom Datenschutz-Beauftragten gemeinsam mit dem Sicherheits-Administrator aufgedeckt werden. Da der Datenschutz-Beauftragte vor der Aufdeckung manuell prüft, ob tatsächlich ein Anfangsverdacht vorliegt, führt dieses Verfahren zu Verzögerungen und ist deswegen nur für den Ausnahmefall vorzusehen.

4.1 Anwendungsbeispiel

Die Wirkungsweise der in Abschnitt 4 vorgestellten Möglichkeiten werden im folgenden an einem einfachen Beispiel gezeigt. Die Darstellung orientiert sich an den in Abb. 5 dargestellten Audit-Daten, die Anhaltspunkte für das Raten des Paßwortes für das Nutzerkonto `sven` enthalten (s. Abb. 5 Zeilen 4-6).

Die Namen der Nutzerkonten, und je nach Infrastruktur auch die Terminalbezeichner, stellen personenbezogene bzw. personenbeziehbare Merkmale dar und sollen im Interesse der Nutzer-Anonymität durch Pseudonyme ersetzt werden. In Abb. 4 spezifizieren die Zeilen 2 und 6 die zu anonymisierenden Anhaltspunkte (`FACILITY` und `EVENT`). Die Merkmals-Kontexte zum Auffinden der Namen der Nutzerkonten sind in den Zeilen 4 und 7, die Merkmals-Kontexte zum Auffinden der Terminalbezeichner in Zeile 3 definiert (`LEFT` und `RIGHT`).

Die Sicherheits-Administratoren möchten allerdings erfahren, welche Nutzerkonten das Ziel von Paßwort-Rate-Angriffen sind. Die Terminalbezeichner möchten die Sicherheits-Administratoren generell nicht erfahren; es reicht ihnen zu wissen, ob Paßwort-Rate-Versuche vom selben Terminal stammen.

```

1: CONTEXT pwlogin THRESHOLD 3

2: FACILITY login EVENT 'FAILED LOGIN on'
3:   LEFT 'on .', RIGHT '' ' TYPE STRING KEEPLEN LINK NORECOVER
4:   LEFT 'FOR .', RIGHT '.', ' TYPE STRING LEN 8 UNLINK RECOVER
5:     CONTEXT pwlogin GROW DEL 0 ADD 1

6: FACILITY PAM_unix EVENT 'login. session opened for user'
7:   LEFT 'opened for user ', RIGHT ' by LOGIN' TYPE STRING LEN 8 UNLINK RECOVER
8:     CONTEXT pwlogin GROW DEL 2 ADD 0

```

Abbildung 4: Beispiel – Konfiguration zum Pseudonymisieren der Audit-Daten in Abb. 5

```

1: Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'tty1' FOR 'sven', Authentication failure
2: Jan 13 17:59:56 oin login[341]: FAILED LOGIN on 'tty1' FOR 'sven', Authentication failure
3: Jan 13 18:00:23 oin PAM_unix[3453]: (login) session opened for user sven by LOGIN(uid=0)
4: Jan 13 19:29:26 oin login[341]: FAILED LOGIN on 'tty2' FOR 'sven', Authentication failure
5: Jan 13 19:29:41 oin login[341]: FAILED LOGIN on 'tty2' FOR 'sven', Authentication failure
6: Jan 13 19:29:58 oin login[341]: FAILED LOGIN on 'tty2' FOR 'sven', Authentication failure

```

Abbildung 5: Beispiel – Login-Audit-Daten, die Paßwort-Rateversuche enthalten

```

1 : Jan  6 12:04:33 oin login[341]: FAILED LOGIN on 'ucSj' FOR 'xVXZrQPu', Authentication failure
2 : Jan 10 16:47:21 oin login[341]: FAILED LOGIN on 'ucSj' FOR 'jFFPmkew', Authentication failure
3 : Jan 12 10:18:42 oin PAM_unix[3453]: (login) session opened for user QWhheUZx by LOGIN(uid=0)
4 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'JfhuOOE1', Authentication failure
5 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'OuGwq8VI', Authentication failure
6 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR '5IkXjc2N', Authentication failure

```

Abbildung 6: Beispiel – Audit-Daten aus Abb. 5 nach der Pseudonymisierung, ohne kryptographisches Material

```

1: Jan  6 12:04:33 oin login[341]: FAILED LOGIN on 'ucSj' FOR 'xVXZrQPu', Authentication failure
2: Jan 10 16:47:21 oin login[341]: FAILED LOGIN on 'ucSj' FOR 'jFFPmkew', Authentication failure
3: Jan 12 10:18:42 oin PAM_unix[3453]: (login) session opened for user QWhheUZx by LOGIN(uid=0)
4: Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'sven', Authentication failure
5: Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'sven', Authentication failure
6: Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'sven', Authentication failure

```

Abbildung 7: Beispiel – pseudonymisierte Login-Audit-Daten aus Abb. 6 mit aufgedeckten Merkmalen, ohne kryptographisches Material

```

1a: Jan 6 12:04:33 oin pseudonymizer: ref=1042626924 nym=xVXzrQPu context=pwlogin label=y6kvQdk recovery=Hs56MmEPUDProZvZU4RwQ:1:LjLe!d!o_WFGuHNo!dhtv1rgZ6E.PHgJmW
1b: Jan 6 12:04:33 oin pseudonymizer: ref=1042626924 nym=xVXzrQPu context=pwlogin label=y6kvQdk recovery=Hs56MmEPUDProZvZU4RwQ:1:LjLe!d!o_WFGuHNo!dhtv1rgZ6E.PHgJmW
1 : Jan 6 12:04:33 oin login[341]: FAILED LOGIN on 'ucsj' FOR 'xVXzrQPu', Authentication failure
2a: Jan 10 16:47:21 oin pseudonymizer: ref=1042626925 nym=jFFPmkew context=pwlogin label=y6kvQdk recovery=hrRf0rezyR1sWwsad6tR00:2:k6P3jV1UFR:PLDedGx3LM3FpxY:QMwJnQ
2b: Jan 10 16:47:21 oin pseudonymizer: ref=1042626925 nym=h0HmHpjXBRdKfI4GgFXkncUA8 origevent=R8YTusxf4CQQtckJvzmrFvhlIE
2 : Jan 10 16:47:21 oin login[341]: FAILED LOGIN on 'ucsj' FOR 'jFFPmkew', Authentication failure
3a: Jan 12 10:18:42 oin pseudonymizer: ref=1042626926 nym=QWhheUzX context=pwlogin label=520eQ8w
3b: Jan 12 10:18:42 oin pseudonymizer: ref=1042626926 nym=QWhheUzX context=pwlogin label=520eQ8w
3 : Jan 12 10:18:42 oin PAM_unix[3453]: (login) session opened for user QWhheUzX by LOGIN(uid=0)
4a: Jan 13 17:59:46 oin pseudonymizer: ref=1042626927 nym=jFhu0E1 context=pwlogin label=hYfrwpY recovery=asiCqeHXI5W6ZACsIlprA:1:xAzRgyLtXQ0MrLy9HoWOWE4phIo:IgBbhW
4b: Jan 13 17:59:46 oin pseudonymizer: ref=1042626927 nym=jFhu0E1 context=pwlogin label=hYfrwpY recovery=asiCqeHXI5W6ZACsIlprA:1:xAzRgyLtXQ0MrLy9HoWOWE4phIo:IgBbhW
4 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'jFhu0E1', Authentication failure
5a: Jan 13 17:59:46 oin pseudonymizer: ref=1042626928 nym=OuGwq8VI context=pwlogin label=hYfrwpY recovery=7pTaXNS!2RJCLsdrVshp7g:2:VVJ0_i5750l_LM2tH9njB773jy8:0_PLYA
5b: Jan 13 17:59:46 oin pseudonymizer: ref=1042626928 nym=OuGwq8VI context=pwlogin label=hYfrwpY recovery=7pTaXNS!2RJCLsdrVshp7g:2:VVJ0_i5750l_LM2tH9njB773jy8:0_PLYA
5 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'OuGwq8VI', Authentication failure
6a: Jan 13 17:59:46 oin pseudonymizer: ref=1042626929 nym=5IkXjc2N context=pwlogin label=hYfrwpY recovery=GKVBMLKuzlGjzvlzHNpRw:3:pZjE6UaLxrolJYq5KwryB:IOZRXXg:_wOGTA
6b: Jan 13 17:59:46 oin pseudonymizer: ref=1042626929 nym=5IkXjc2N context=pwlogin label=hYfrwpY recovery=GKVBMLKuzlGjzvlzHNpRw:3:pZjE6UaLxrolJYq5KwryB:IOZRXXg:_wOGTA
6 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR '5IkXjc2N', Authentication failure

```

Abbildung 8: Beispiel – pseudonymisierte Audit-Daten aus Abb. 6, mit kryptographischem Material

```

1a: Jan 6 12:04:33 oin pseudonymizer: ref=1042626924 nym=xVXzrQPu context=pwlogin label=y6kvQdk recovery=Hs56MmEPUDProZvZU4RwQ:1:LjLe!d!o_WFGuHNo!dhtv1rgZ6E.PHgJmW
1b: Jan 6 12:04:33 oin pseudonymizer: ref=1042626924 nym=xVXzrQPu context=pwlogin label=y6kvQdk recovery=Hs56MmEPUDProZvZU4RwQ:1:LjLe!d!o_WFGuHNo!dhtv1rgZ6E.PHgJmW
1 : Jan 6 12:04:33 oin login[341]: FAILED LOGIN on 'ucsj' FOR 'xVXzrQPu', Authentication failure
2a: Jan 10 16:47:21 oin pseudonymizer: ref=1042626925 nym=jFFPmkew context=pwlogin label=y6kvQdk recovery=hrRf0rezyR1sWwsad6tR00:2:k6P3jV1UFR:PLDedGx3LM3FpxY:QMwJnQ
2b: Jan 10 16:47:21 oin pseudonymizer: ref=1042626925 nym=h0HmHpjXBRdKfI4GgFXkncUA8 origevent=R8YTusxf4CQQtckJvzmrFvhlIE
2 : Jan 10 16:47:21 oin login[341]: FAILED LOGIN on 'ucsj' FOR 'jFFPmkew', Authentication failure
3a: Jan 12 10:18:42 oin pseudonymizer: ref=1042626926 nym=QWhheUzX context=pwlogin label=520eQ8w
3b: Jan 12 10:18:42 oin pseudonymizer: ref=1042626926 nym=QWhheUzX context=pwlogin label=520eQ8w
3 : Jan 12 10:18:42 oin PAM_unix[3453]: (login) session opened for user QWhheUzX by LOGIN(uid=0)
4a: Jan 13 17:59:46 oin pseudonymizer: ref=1042626927 nym=jFhu0E1 context=pwlogin label=hYfrwpY recovery=asiCqeHXI5W6ZACsIlprA:1:xAzRgyLtXQ0MrLy9HoWOWE4phIo:IgBbhW
4b: Jan 13 17:59:46 oin pseudonymizer: ref=1042626927 nym=jFhu0E1 context=pwlogin label=hYfrwpY recovery=asiCqeHXI5W6ZACsIlprA:1:xAzRgyLtXQ0MrLy9HoWOWE4phIo:IgBbhW
4 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'jFhu0E1', Authentication failure
5a: Jan 13 17:59:46 oin pseudonymizer: ref=1042626928 nym=OuGwq8VI context=pwlogin label=hYfrwpY recovery=7pTaXNS!2RJCLsdrVshp7g:2:VVJ0_i5750l_LM2tH9njB773jy8:0_PLYA
5b: Jan 13 17:59:46 oin pseudonymizer: ref=1042626928 nym=OuGwq8VI context=pwlogin label=hYfrwpY recovery=7pTaXNS!2RJCLsdrVshp7g:2:VVJ0_i5750l_LM2tH9njB773jy8:0_PLYA
5 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR 'OuGwq8VI', Authentication failure
6a: Jan 13 17:59:46 oin pseudonymizer: ref=1042626929 nym=5IkXjc2N context=pwlogin label=hYfrwpY recovery=GKVBMLKuzlGjzvlzHNpRw:3:pZjE6UaLxrolJYq5KwryB:IOZRXXg:_wOGTA
6b: Jan 13 17:59:46 oin pseudonymizer: ref=1042626929 nym=5IkXjc2N context=pwlogin label=hYfrwpY recovery=GKVBMLKuzlGjzvlzHNpRw:3:pZjE6UaLxrolJYq5KwryB:IOZRXXg:_wOGTA
6 : Jan 13 17:59:46 oin login[341]: FAILED LOGIN on 'PIHq' FOR '5IkXjc2N', Authentication failure

```

Abbildung 9: Beispiel – reidentifizierte Audit-Daten aus Abb. 7, mit kryptographischem Material

Der Anfangsverdacht für einen Paßwort-Rate-Angriff `pwbrute` sei hier als das dreimalige Auftreten des Anhaltspunkts “Anmeldeversuch fehlgeschlagen”. Der Anhaltspunkt “Anmeldeversuch erfolgreich” soll den Anfangsverdacht `pwbrute` entkräften, sofern weniger als drei fehlgeschlagene Anmeldeversuche vorangingen.

Dementsprechend sollen die Terminalbezeichner (s. Abb. 4, Zeile 3) durch ebenso lange (`STRING KEEPLEN`) und verkettbare (`LINK`) Pseudonyme ersetzt werden, die sich nicht geordnet aufdecken lassen (`NORECOVER`). Da die Pseudonyme nicht aufdeckbar sein sollen, haben sie keinen Einfluß auf den Anfangsverdacht.

Die Namen der Nutzerkonten (s. Abb. 4, Zeile 4) in den Anhaltspunkten “Anmeldeversuch fehlgeschlagen” sollen immer durch aufdeckbare (`RECOVER`) Pseudonyme der Länge 8 (`STRING LEN 8`) ersetzt werden, damit die Pseudonymlänge nicht bereits Rückschlüsse auf den Namen des Nutzerkontos zuläßt. Die Pseudonyme seien unverkettbar (`UNLINK`), damit aufgedeckte Pseudonyme keine Rückschlüsse auf Anmeldevorgänge zulassen, die nicht im Zusammenhang mit einem Paßwort-Rate-Angriff stehen. Das Anhaltspunkts-Merkmal wird dem Anfangsverdachts-Kontext `pwbrute` zugeordnet (s. `CONTEXT` in Abb. 4, Zeile 5). Dabei ist jedes Auftreten des Anhaltspunkts-Merkmals signifikant (`GROW`) und verstärkt den Anfangsverdacht mit dem Gewicht eins (`ADD`). Der Schwellenwert, der zum Anfangsverdachts-Kontext `pwbrute` gehört, sei drei (s. `THRESHOLD` in Abb. 4, Zeile 1). Dementsprechend wird der Anfangsverdacht erfüllt und die Pseudonyme der Anhaltspunkts-Merkmale sind aufdeckbar, wenn der Anhaltspunkt “Anmeldeversuch fehlgeschlagen” dreimal in Folge auftritt. Analog ist der Einfluß des Merkmals Nutzerkonto-Name beim Anhaltspunkt “Anmeldeversuch erfolgreich” entgegengesetzt. Jedes Auftreten dieses Anhaltspunkts-Merkmals soll den Anfangsverdacht mit dem Gewicht zwei abschwächen (s. `DEL` in Abb. 4, Zeile 8).

Das Resultat der Pseudonymisierung der Beispiel-Audit-Daten in Abb. 5 ist in Abb. 6 dargestellt. Während Abb. 6 nur die pseudonymisierten Audit-Daten zeigt, ist das zugehörige vom `pseudonymizer` erzeugte kryptographische Material in Abb. 8 zu sehen. Mit Hilfe des kryptographischen Materials ist der `reidentifizier` in der Lage, die Pseudonyme der Anhaltspunkte aufzudecken, die den Anfangsverdacht `pwbrute` erfüllen (s. Abb. 7, Zeilen 4-6). Das kryptographische Material aufgedeckter Pseudonyme entfernt der `reidentifizier` automatisch (vgl. Abb. 8 und Abb. 9, Zeilen 4a, 5a und 6a).

Anhand des kryptographischen Materials wird deutlich, daß die Pseudonyme der Nutzerkonten-Namen faktisch nicht unverkettbar sind, wie es zunächst bei alleiniger Betrachtung der pseudonymisierten Audit-Daten in Abb. 6 scheint. Vielmehr sind jene Pseudonyme vermöge des `label`-Werts `hYfrwpY` im kryptographischen Material verkettbar, welche zum erfüllten Anfangsverdacht gehören (s. Abb. 8, Zeilen 4a, 5a und 6a). Sie sind allerdings nicht mit den unaufgedeckten Pseudonymen verkettbar (s. Abb. 8, Zeilen 1a, 2a und 3a).

Tabelle 1: Maximales Audit-Datenvolumen

	Anzahl Audit-Datensätze pro Stunde	pro Sekunde
<i>Web-Server</i>	33506	9.31
<i>Syslog</i>	4956	1.38
Σ	38462	10.68

Tabelle 2: Performanz der Geheimnisteilung

Schwellenwert	Initialisierungen/s	Anteile/s	Interpolationen/s
≤ 10	> 1080	> 33280	> 7460
≤ 100	> 140	> 1620	> 620
≤ 1000	> 15	> 90	> 60

5 Laufzeitverhalten

Der Pseudonymisierer läuft ununterbrochen im System mit und verarbeitet die Audit-Daten sofort nach ihrer Erhebung. Dabei ist es wichtig, daß er hinreichend schnell arbeitet und ein Daten-Rückstau lediglich temporärer Natur ist. Der Reidentifizierer hingegen muß keine so strengen Laufzeitanforderungen erfüllen, da er nur vereinzelt und gezielt zum Einsatz kommt.

Um ein realistisches Maß für übliche Audit-Datenvolumina zu erhalten, wurden die *Syslog*- und *Web-Server*-Audit-Daten eines zentralen Servers am Zentrum für Kommunikation und Informationsverarbeitung der Universität Dortmund ausgewertet. Die betrachtete Solaris SUN Ultra Enterprise 4000 Maschine verfügt über sechs Ultra SPARC 168MHz CPUs, 3GB RAM und drei Platten-Arrays mit insgesamt 396GB sowie eine 100Mbps Netzwerkkarte. Während der Arbeitszeit sind von den 1050 registrierten Nutzern durchschnittlich 25 Nutzer simultan auf der Maschine tätig. Die Maschine trägt 37 weltweit erreichbare *Web-Server*, einen *FTP-Server* mit monatlich 112000 Transfers und einem Transfervolumen von 12GB, sowie *Email-Dienste* im Umfang von monatlich 45000 Emails. Die beobachteten maximalen stündlichen Audit-Datenaufkommen sind in Tabelle 1 dargestellt [5].

Laufzeitmessungen des Pseudonymisierers wurden bei einer Schlüssel- und Zahlenlänge von 128 Bit auf einem OpenBSD-Rechner mit einer Pentium III 650MHz CPU mit 256MB RAM und einer 100Mbps Netzwerkkarte durchgeführt. Die Mikrobenchmarks der Kryptoprimitive zeigen, daß diese mehr als hinreichend schnell sind. Die Meßergebnisse sind in Tabelle 2 zu finden. Umfangreiche Makrobenchmarks des Pseudonymisierers zeigen eine Verarbeitungsgeschwindigkeit zwischen 1060 und 70 Audit-Datensätzen pro Sekunde. Die Verarbeitungsgeschwindigkeit ist von verschiedenen Parametern abhängig, z.B. der Anzahl der zu pseudonymisierenden Personenbezüge je Datensatz. Es wird jedoch deutlich, daß selbst für ungünstige Parameterwerte der Durchsatz des Pseudonymisierers deutlich über den maximalen anfallenden Audit-Datenvolumina liegt [5].

6 Danksagung

Besonderer Dank gebührt Sven Bursch, Kai Grundmann und Dennis Real für ihr Engagement bei der Implementierung von *Pseudo/CoRe*.

Literaturverzeichnis

- [1] Joachim Biskup and Ulrich Flegel. On pseudonymization of audit data for intrusion detection. In Hannes Federrath, editor, *Proceedings of the international Workshop on Design Issues in Anonymity and Unobservability*, number 2009 in LNCS, pages 161–180, Berkeley, California, July 2000. ICSI, Springer.
- [2] Joachim Biskup and Ulrich Flegel. Transaction-based pseudonyms in audit data for privacy respecting intrusion detection. In Hervé Debar, Ludovic Mé, and S. Felix Wu, editors, *Proceedings of the Third International Symposium on Recent Advances in Intrusion Detection (RAID 2000)*, number 1907 in LNCS, pages 28–48, Toulouse, France, October 2000. Springer.
- [3] Joachim Biskup and Ulrich Flegel. Threshold-based identity recovery for privacy enhanced applications. In Sushil Jajodia and Pierangela Samarati, editors, *Proceedings of the 7th ACM Conference on Computer and Communications Security*, pages 71–79, Athens, Greece, November 2000. ACM SIGSAC, ACM Press.
- [4] Joachim Biskup and Ulrich Flegel. Ausgleich von Datenschutz und Überwachung mit technischer Zweckbindung am Beispiel eines Pseudonymisierers (in German). In Sigrid Schubert, Bernd Reusch, and Norbert Jesse, editors, *Informatik bewegt, Proceedings of the 32nd Annual GI Conference on Informatik (Informatik 2002) (in German)*, number P-19 in Lecture Notes in Informatics, pages 488–494, Dortmund, Germany, October 2002. Gesellschaft für Informatik e.V.(GI), Köllen Verlag.
- [5] Ulrich Flegel. Pseudonymizing Unix log files. Technical report, Dept. of Computer Science, Chair VI Information Systems and Security, University of Dortmund, D-44221 Dortmund, May 2002. <http://ls6-www.cs.uni-dortmund.de/issi/archive/literature/2002/Flegel:2002a.ps.gz>.
- [6] Ulrich Flegel. Anonyme Audit-Daten im Überblick (in German). *Datenschutz und Datensicherheit*, 27(5):278–281, May 2003.
- [7] Alexander Roßnagel and Philip Scholz. Datenschutz durch Anonymität und Pseudonymität (in German). *Zeitschrift für Informations-, Telekommunikations- und Medienrecht (MMR)*, 2000(12):721–732, 2000.
- [8] Stefan Jaeger. Verbotene Protokolle (in German). *Zeitschrift für Kommunikations- und EDV-Sicherheit (KES)*, 2000(5):6–12, 2000.
- [9] Michael Sobirey. *Datenschutzorientiertes Intrusion Detection (in German)*. DuD-Fachbeiträge. Vieweg, 1999.

-
- [10] Herbert Fiedler. Der Staat im Cyberspace (in German). *Informatik Spektrum*, 24(5):309–314, 2001.
- [11] Alexander Roßnagel. Freiheit im Cyberspace (in German). *Informatik Spektrum*, 25(1):33–38, 2002.
- [12] Herbert Fiedler. Cyber-libertär (in German). *Informatik Spektrum*, 25(3):215–219, 2002.
- [13] Claudia Eckert and Alexander Pircher. Internet anonymity: Problems and solutions. In Michel Dupuy and Pierre Paradinas, editors, *Proceedings of the IFIP TC11 16th International Conference on Information Security (IFIP/Sec'01)*, pages 35–50, Paris, France, June 2001. IFIP, Kluwer Academic Publishers.
- [14] Emilie Lundin and Erland Jonsson. Anomaly-based intrusion detection: privacy concerns and other problems. *Computer Networks*, 34(4):623–640, October 2000.
- [15] webwasher.com AG. Den Überblick behalten, reporting mit WebWasherEE. http://www.webwasher.com/product_pdf/deutsch/Produktblatt_Reporting.pdf, January 2003.
- [16] Simone Fischer-Hübner. *IDA (Intrusion Detection and Avoidance System): Ein einbruchsentdeckendes und einbruchsvermeidendes System (in German)*. Reihe Informatik. Shaker, 1993.
- [17] Michael Meier and Thomas Holz. Sicheres Schlüsselmanagement für verteilte Intrusion-Detection-Systeme (in German). In Patrick Horster, editor, *Systemicherheit*, DuD-Fachbeiträge, pages 275–286, Bremen, Germany, March 2000. GI-2.5.3, ITG-6.2, ÖCG/ACS, TeleTrusT, Vieweg.
- [18] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.
- [19] *OpenSSL cryptographic library*, December 2001. <http://www.openssl.org/docs/crypto/crypto.html>.
- [20] Torbjörn Granlund. *The GNU Multiple Precision Arithmetic Library*. GNU, 3.1.1 edition, September 2000. <http://www.gnu.org/manual/gmp/index.html>.
- [21] Victor Shoup. NTL: A library for doing number theory. <http://www.shoup.net/ntl/>, 2003.
- [22] *OpenSSL SSL/TLS library*, December 2001. <http://www.openssl.org/docs/ssl/ssl.html>.