*G. Wolf, H. Federrath, A. Pfitzmann, A. Schill: Endbenutzer- und Entwicklerunterstützung bei der Durchsetzung mehrseitiger Sicherheit. In: P. Horster (Hrsg.): Sicherheitsinfrastrukturen – Grundlagen, Realisierungen, Rechtliche Aspekte, Anwendungen. Proceedings des Workshops, Hamburg, 9.-10. März 1999, Vieweg, 17-30*

# Endbenutzer- und Entwicklerunterstützung bei der Durchsetzung mehrseitiger Sicherheit

Gritta Wolf, Hannes Federrath, Andreas Pfitzmann, Alexander Schill

TU Dresden, Fakultät Informatik, 01062 Dresden
E-Mail: {g.wolf, federrath, pfitza, schill}@inf.tu-dresden.de

## Zusammenfassung

Ausgehend von den Begriffen mehrseitige Sicherheit und Architektur wird geklärt, welche Anforderungen eine Sicherheitsarchitektur für die Unterstützung von Endbenutzern und Entwicklern erfüllen sollte. Im Zusammenhang mit der Realisierung mehrseitiger Sicherheit besitzt die Formulierung, Durchsetzung und ggf. Aushandlung von Schutzzielen bzw. Sicherheitseigenschaften große Bedeutung. Sowohl für den Nutzer als auch für den Entwickler mehrseitig sicherer Kommunikationssysteme sind unterstützende Funktionen notwendig, die exemplarisch beschrieben werden. Am Beispiel einer Architektur für mehrseitige Sicherheit (SSONET, Sicherheit und Schutz in offenen Datennetzen) wird die Unterstützung demonstriert.

## 1 Sicherheit und mehrseitige Sicherheit

In den letzten Jahren ist zu beobachten, daß eine Vielzahl von Kommunikationsanwendungen für offene Systeme nach und nach eine Erweiterung um Sicherheitsfunktionen erfährt und alte Sicherheitsanwendungen immer komfortabler werden. Beispiele sind E-Mail-Clients, Secure Shell, PGP. Leider präsentiert sich dem *Endbenutzer* die angebotene Sicherheitsfunktionalität je nach Anwendung auf sehr unterschiedliche Weise; Kommunikationsschlüssel können teilweise nicht anwendungsübergreifend genutzt werden und verbindliche Zusicherungen über die erbrachte bzw. erreichte Sicherheit fehlen meist. Auch dem *Entwickler* von Kommunikationsanwendungen ist es schwer möglich, sich auf standardisierte Sicherheitsschnittstellen zu stützen, wenn er sich überhaupt dazu durchgerungen hat, neben den Primärfunktionen der Anwendungen (d.h. den eigentlichen Aufgaben, die eine Anwendung erfüllen soll) auch Sicherheitsmerkmale anzubieten. Es besteht also im Bereich Sicherheit sowohl Bedarf an Endbenutzerunterstützung als auch an Entwicklerunterstützung.

Unter Sicherheit eines Kommunikationssystems soll die Durchsetzung von Schutzzielen (Vertraulichkeit, Integrität, Verfügbarkeit) trotz Vorhandensein intelligenter Angreifer verstanden werden. Werden bei der Durchsetzung dieser Schutzziele die Sicherheitswünsche aller an einer Kommunikation beteiligten Instanzen berücksichtigt und möglicherweise entstehende Schutzkonflikte erkannt und ausgetragen, spricht man von *mehrseitiger Sicherheit*.

Es lassen sich die folgenden Phasen der Aushandlung einer mehrseitig sicheren Kommunikationsbeziehung unterscheiden (hier am Beispiel einer Kommunikation zweier Teilnehmer A und B, siehe Bild 1):

1. **A:** Explizite Formulierung von individuellen Schutzinteressen (Konfiguration)
2. **A→B:** Übermittlung des Kommunikationswunsches unter Einbeziehung der Schutzinteressen
3. **B:** Explizite Formulierung von individuellen Schutzinteressen (Konfiguration), möglicherweise Rekonfiguration unter Einbeziehung der Interessen von A (Aushandlung). Die Konfigurationsphase für B kann auch vor der Übermittlung des Kommunikationswunsches stattgefunden haben, berücksichtigt dann natürlich nicht unmittelbar die Interessen von A.

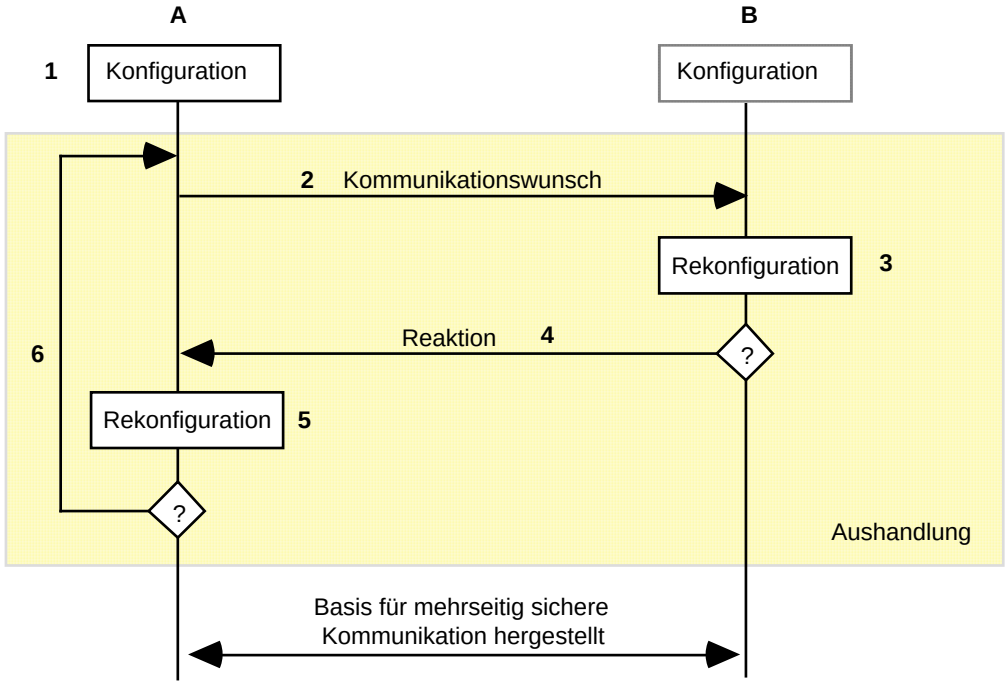


**Bild 1.** Konfiguration und Aushandlung als Elementarphasen beim Zustandekommen einer mehrseitig sicheren Kommunikationsbeziehung

4. **B→A:** Reaktion von B auf den Kommunikationswunsch von A unter Einbeziehung der eigenen Schutzinteressen. Ggf. ist B bereits mit den Sicherheitswünschen von A einverstanden und dem Aufbau einer mehrseitig sicheren Kommunikationsverbindung steht nichts mehr im Wege.
5. **A:** Ggf. Rekonfiguration der eigenen Schutzinteressen unter Berücksichtigung derer von B (Aushandlung)
6. **A:** Danach kann eine neue Kommunikationsbeziehung ausgehandelt werden (wobei natürlich die Aushandlungsergebnisse der Phasen 3 und 5 der vorherigen Kommunikationen die Konfiguration der folgenden beeinflussen sollten) oder es sind die Grundlagen für eine mehrseitig sichere Kommunikationsbeziehung zwischen A und B geschaffen.

# 2 Sicherheitsarchitekturen

Wenn es möglich sein soll, Sicherheitsfunktionen system- und anwendungsübergreifend nutzbar zu machen, werden Grundfunktionen benötigt, die allen Systemkomponenten

(Betriebssystem, Netz, Anwendungen) zur Verfügung stehen. Diese Funktionen werden von einer Sicherheitsarchitektur erbracht.

Der Architekturbegriff umfaßt allgemein die Beschreibung des *Zwecks* und die Beschreibung einer dem Zweck angepaßten *Form*. Neben dem Zweck oder auch der Funktionalität einer Architektur kann unter Form auch die Beschreibung der Integration der Architektur in bestehende Systeme und die Beschreibung der umgebenden Infrastruktur gefaßt werden.

In der Informatik sind im engeren Sinn Systemarchitekturen von Bedeutung. Systeme können Rechner, Rechnernetze, aber auch Programmiersprachen sein. Eine Systemarchitektur beschreibt sowohl in konkreter als auch in abstrakter Form alle wesentlichen Komponenten für den Bau einer Systemlösung. Die Komponenten müssen in ihrem Zusammenwirken die gewünschte Funktionalität der Lösung erbringen. Architekturen beziehen sich auf Lösungen für eine bestimmte Problemstellung oder einen bestimmten Anwendungsbereich. Architekturkomponenten können in Hardware, Software oder der Kombination von beidem implementiert sein.

Unter Sicherheitsarchitektur soll im folgenden ebenfalls eine Systemarchitektur verstanden werden, die es erlaubt, insbesondere die Sicherheits- und Zuverlässigkeitsaspekte eines Rechners bzw. Rechnernetzes zu verstehen und zu beschreiben. Sicherheitsarchitekturen sollen in geordneter und nachvollziehbarer Weise ein Rahmenkonzept für die Integration systemübergreifender Sicherheitsfunktionen in informationstechnische Systeme schaffen. Es muß unterschieden werden, für welche Problemstellungen Sicherheitsfunktionalität realisiert werden soll, z.B. für die

- Sicherung der Kommunikation,
- Sicherung eines lokalen Systems und/oder
- Erfüllung organisatorischer Rahmenbedingungen.

Organisatorische Rahmenbedingungen wie zum Beispiel Schutz gegen Folgen von Naturkatastrophen, Abstrahlsicherheit (physikalische Sicherheit), Richtlinien zu Nutzerberechtigungen (administrative Sicherheit) und Zugangsberechtigungen (Personalsicherheit) [vgl. NoHe_90] gehen über die informationstechnischen Aspekte einer Sicherheitsarchitektur hinaus und werden im Weiteren nicht betrachtet.

## 2.1 Ansätze für Sicherheitsarchitekturen

Sicherheitsarchitekturen umfassen aufgrund ihrer systemübergreifenden Konzeption verschiedene Systembereiche: Betriebssysteme, Schnittstellen, (Krypto-)Bibliotheken, Rechnernetze (insbesondere verteilte Systeme) sowie Anwendungen.

| Systembereich | Typische Vertreter [Literatur] |
|---|---|
| Betriebssysteme | BirliX [HäKK_93] |
| | Windows NT [Cust_95] |
| Schnittstellen | GSS-API [Linn_90] |
| | Microsoft Crypto API [Wiew_96] |
| (Krypto)-Bibliotheken | CM++ [BaBl_96] |
| | Cryptix [Cryp] |
| | LiSA [BKMM_96] |
| | RSAREF [RSA] |
| | SecuDE [Secu] |
| | SSLeay [SSL] |
| Rechnernetze, Verteilte Systeme | Kerberos [StNS_88] |
| | DCE [Schi_97] |
| | CORBA [OMG_95] |
| | TINA/CrySTINA [SBHS_97] |
| Anwendungen | Electronic Mail, z.B. Pretty Good Privacy [PGP] |
| | Electronic Commerce, z.B. SEMPER [Waid_96] |
| | Zahlungssysteme, z.B. SET [SET_97] |
| Systemübergreifende Ansätze | CDSA [CDSA_96] |
| | CISS [MPSC_93] |
| | DSSA [GGKL_89] |
| | PLASMA [Kran_96] |
| | REMO [REMO_93] |
| | SSONET [SSONET] |

**Tabelle 1.** Ansätze für Sicherheitsarchitekturen (Auswahl)

Die existierenden Ansätze für Sicherheitsarchitekturen sind jeweils stark von dem Systembereich geprägt, dem sie entstammen. Die Tabelle 1 gibt eine Übersicht über ausgewählte Ansätze.

Die systemübergreifenden Ansätze für Sicherheitsarchitekturen betrachten im wesentlichen die Integration von Sicherheitspolitiken sowie Aushandlungs- und Auflösungsstrategien für Sicherheitskonflikte.

Defizite finden sich vor allem in der Systematisierung der Ansätze, der praktischen Umsetzung sowie der Unterstützung weiterreichender Schutzziele wie Anonymität, Unbeobachtbarkeit, Unverkettbarkeit und Pseudonymität.

Ein entscheidender Aspekt für die praktische Relevanz neuer Sicherheitskonzepte ist die technische und organisatorische Abstimmung der Entwicklung der Sicherheitsarchitektur mit der Entwicklung von zukünftigen Netzarchitekturen allgemein, wie z.B. von CORBA als Middleware-Architektur und der Telecommunications Information Networking Architecture (TINA, [DuNI_95]) für Telekommunikationsnetze. Gerade der Mehrseitigkeitsaspekt im Geschäftsmodell neuer Telekommunikationsnetze erfordert das Konzept mehrseitiger Sicherheit.

## 2.2 Anforderungen: Was sollen Sicherheitsarchitekturen leisten?

Im folgenden wird eine Auswahl an Anforderungen und Funktionen angegeben, die Sicherheitsarchitekturen erfüllen sollen. In Anlehnung an CISS [MPSC_93, S.87] und [NoHe_90, S.9ff] kann die Funktionalität von Sicherheitsarchitekturen wie in Tabelle 2 dargestellt untergliedert werden.

Alle 4 Bereiche können Bestandteil einer umfassenden Sicherheitsarchitektur sein, wobei sich die unter 1. fallenden Rahmenbedingungen mit der Umgebung der zu sichernden Rechnersysteme beschäftigen, 2. mit der Sicherheit eines lokalen Rechnersystems und 3. mit Sicherheit bei der Kommunikation. Nur implizit erwähnt wird die Verfügbarkeit. 4. weist direkt auf die Infrastruktur um die Sicherheitsarchitektur hin.

Für die Realisierung mehrseitiger Sicherheit muß 1. und 2. gelöst werden (bzw. wird vorausgesetzt). Unter 3. können Primäraspekte (Anwendung von Sicherheitsmechanismen zur Erreichung von ausgehandelten Schutzzielen) und unter 4. Sekundäraspekte (Infrastruktur wie Schlüsselserver, TTPs etc.) mehrseitiger Sicherheit eingeordnet werden.

Für die Durchsetzung mehrseitiger Sicherheit müssen insbesondere Endbenutzer in die Lage versetzt werden, ihre Sicherheitsinteressen zu formulieren. Deshalb sollte neben der Systemunterstützung für die am Design- und Entwicklungsprozeß beteiligten Rollen besonderer Wert auf die Gestaltung der Benutzeroberfläche gelegt werden. In [ElNa_87] werden folgende Bewertungskriterien für Benutzerschnittstellen genannt, die auch für die Gestaltung mehrseitig sicherer Systeme gelten können:

- Einfachheit (simplicity),
- Effizienz (efficiency),
- Nutzbarkeit/Benutzerfreundlichkeit (usability).

Im folgenden werden einige detaillierte Konzepte zur Erfüllung dieser Kriterien im Rahmen einer Architektur für mehrseitige Sicherheit vorgeschlagen.

| | |
|---|---|
| **1.** | **organisatorische und physikalische Rahmenbedingungen [NoHe_90]** |
| • | physikalische Sicherheit |
| - | Schutz gegen Folgen von Naturkatastrophen |
| - | Abstrahlsicherheit |
| • | administrative Sicherheit |
| - | Richtlinien der Unternehmensverwaltung dafür, welche Nutzer die Berechtigungen besitzen, mit welchen Systemen was zu tun |
| • | Personalsicherheit |
| - | Firmenausweise, |
| - | Zugangsberechtigungen, etc. |
| **2.** | **Endsystemsicherheit** |
| • | lokale Datensicherheit |
| - | Entitätsauthentifikation, |
| - | Vertraulichkeit von Daten, |
| - | Integrität von Daten, |
| - | Zugriffskontrolle, |
| - | Non-repudiation |
| • | Softwaresicherheit und Prozeßsicherheit [MPSC_93] |
| - | Software-Authentizität und -Integrität (z.B. gegen Viren, trojanische Pferde, etc.) |
| - | Sicherheit von Betriebssystemen |
| - | Implementation von speicherlosen Subsystemen |
| • | Hardwaresicherheit |
| - | Hardware-Integrität |
| **3.** | **Kommunikationssicherheit [MPSC_93]** |
| - | Entitätsauthentifikation, |
| - | Vertraulichkeit von Daten, |
| - | Integrität von Daten, |
| - | Zugriffskontrolle, |
| - | Verbindlichkeit (Non-repudiation), |
| - | Schutz vor Subliminal Channels, |
| - | Schutz vor Denial of Service, |
| - | Sichere Gruppenkommunikation, |
| - | Anonyme Kommunikation |
| **4.** | **Sicherheitsmanagement [MPSC_93]** |
| - | Schlüsselerzeugung, -speicherung und -verteilung |
| - | Logging und Auditing von sicherheitsrelevanten Ereignissen |
| - | Security Recovery |
| - | Notariatsservice |

**Tabelle 2.** Anforderungskatalog an Sicherheitsarchitekturen

# 3 Akteursunterstützung in Sicherheitsarchitekturen

Eine umfassende Sicherheitsarchitektur nützt nichts, wenn die Benutzer nicht damit umgehen können. Am Prozeß von der Erstellung einer Sicherheitsarchitektur bis zur Nutzung gesicherter Anwendungen sind verschiedene Personengruppen bzw. Rollen beteiligt:

1. Entwickler der Sicherheitsarchitektur bzw. Sicherheitsexperten,
2. Anwendungsentwickler, die die Dienste der Sicherheitsarchitektur nutzen und in Anwendungen integrieren,
3. Systemadministratoren, die Einstellungen für ihren speziellen System- oder Organisationsbereich vornehmen oder als persönlicher Berater eines Endbenutzers fungieren,

4. Endbenutzer, die die Anwendungen nutzen.

Jede Rolle erfüllt schrittweise die notwendigen Aufgaben auf dem Weg zur Nutzung einer gesicherten Anwendung. Die Übergänge zwischen den einzelnen Rollen und ihren Aufgaben im Design- und Nutzungsprozeß sind teilweise fließend. Die Aufgaben in den Rollen des Systemadministrators und des Endbenutzers können bei entsprechendem Wissen z.B. von der gleichen Person durchgeführt werden, ebenso wie die Rollen des Sicherheitsarchitektur- und Anwendungsentwicklers oder des Anwendungsentwicklers und Systemadministrators zusammenfallen können.

Es wird diskutiert, bei welchen Tätigkeiten eine Rolle durch welche Konzepte unterstützt wird und durch die Erfüllung welcher Aufgaben eine Rolle ihren nachfolgenden Unterstützung liefert.

Generell nützliche Grundkonzepte zur Unterstützung des Entwicklungs- und Nutzungsprozesses sind u.a. Abstraktion, Information über Funktionalität und Rahmenbedingungen, geeignete Schnittstellengestaltung, Standardvorgaben, Datenverwaltung, Automation von Prozessen und Fehlermeldungen.

Zunächst muß Nutzern klargemacht werden, welche Funktionalität ihnen die Architektur bietet. Ein geeigneter Mechanismus, Funktionalität Nutzern mit unterschiedlichem Wissensstand – also Experten und insbesondere Laien – nahezubringen, ist die *Abstraktion*. Es wird in allgemeiner Form erklärt, welche Dienste die Architektur erbringt bzw. erbringen kann, wobei von speziellen Details und Einzelheiten abstrahiert wird.

Die Fähigkeit eines Nutzers, seine eigenen Sicherheitsinteressen zu formulieren, hängt jedoch vom Wissen des Nutzers z.B. über die Architektur, Eigenschaften von Sicherheitsmechanismen und möglichen Angreifern ab. Der Grad der eigenverantwortlichen Formulierung von Sicherheitsinteressen durch Nichtexperten kann also nur durch ihre *Informiertheit* erhöht werden. Die Information von Nutzern einer Architektur für mehrseitige Sicherheit geht weit über übliche Hilfetexte hinaus. Es müssen Informationen u.a. über die Leistungsfähigkeit und Anwendbarkeit von Sicherheitsmechanismen, über mögliche Angreifer bei der verteilten Kommunikation im Netz und über Auswirkungen der Nutzung von Sicherheitsmechanismen (evtl. Performanceverluste, Rechtsgrundlagen) bereitstehen und geeignet aufbereitet werden. Zur nutzeradäquaten Aufbereitung der Informationen steht u.a. das Hilfsmittel der Abstraktion zur Verfügung.

Aber auch bei guter Nutzerinformation setzt sich die notwendige Nutzerunterstützung weiter fort. Wichtig für den Konfigurierungsvorgang ist einerseits die nicht nur ergonomische, sondern auch funktionale *Gestaltung der Nutzerschnittstelle* (z.B. Verhinderung nicht plausibler Einstellungen), andererseits auch ein Test der getroffenen Einstellungen und die Ausgabe entsprechender *Nutzerhinweise im Fehlerfall*. Inhaltliche Kriterien hierfür sind z.B. geeignete Schutzzielkombinationen, die Nutzbarkeit von Sicherheitsmechanismen für performancekritische (Echtzeit-)Anwendungen, etc.

Trotz der inzwischen relativ hohen Sensibilisierung für Sicherheitsprobleme und oben genannter Konzepte für die Nutzerunterstützung wird oft kritisch hinterfragt, ob Nutzer denn tatsächlich dazu motiviert werden können, sich mit den Sicherheitsfragen und -problemen ihres Systems zu beschäftigen und eine Anzahl von Einstellungen vorzunehmen. Durch *Standardvorgaben* für möglichst viele Teilbereiche der Konfigurierung der Sicherheitsarchitektur kann dieses Problem zumindest verringert werden. Solche Standardvorgaben sind zum Beispiel sinnvoll für die Einstellungen von Mechanismendetails (Schlüssellängen,

Rundenzahlen etc.), die Bewertung von Mechanismen nach dem Grad der erreichbaren Sicherheit und eine dementsprechende Ordnung als Präferenzliste. Aus Anwendungssicht kann bereits eine Auswahl der für die spezifische Anwendung geeigneten Sicherheitsmechanismen vorgegeben werden. Diese Aufgabe kann z.B. durch den Systemadministrator oder sogar den Anwendungsentwickler vorgenommen werden. Solche Standardvorgaben dienen auch als Grundlage zur *Automation* von Teilprozessen.

Ein sekundärer, jedoch trotzdem nicht zu vernachlässigender Aspekt der Nutzerunterstützung ist die *Verwaltung* von Konfigurationsdaten und Standardwerten. Es müssen nicht nur die Einstellungen des einzelnen Endbenutzers, sondern zum Teil auch die ausgehandelten Kommunikationsbasen gespeichert werden. Standardvorgaben sollten aufbewahrt werden, um das System auch nach möglichen Nutzermodifikationen wieder in den Ausgangszustand versetzen zu können.

Im folgenden werden auf der Basis unserer Erfahrungen im Projekt SSONET [SSONET] Konzepte vorgestellt und diskutiert, die dazu beitragen, die am Entwicklungs- und Nutzungsprozeß von Sicherheitsarchitekturen Beteiligten zu unterstützen. Einen Überblick über SSONET gibt Bild 2.

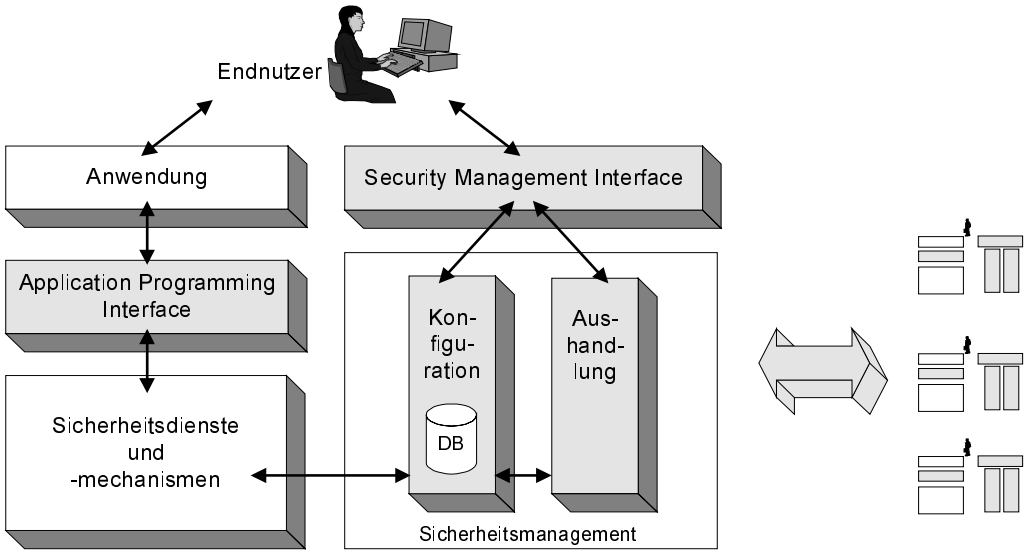


**Bild 2.** Die SSONET-Sicherheitsarchitektur

Die SSONET-Sicherheitsarchitektur bietet eine Sicherheitsmanagementschnittstelle für Endbenutzer und ein API zur Integration von Sicherheitsmechanismen für Anwendungsentwickler. Darüber hinaus ist eine Konfigurationsdatenbank und – zur Umsetzung mehrseitiger Sicherheit – eine Aushandlungskomponente enthalten. Beim Verbindungsaufbau zwischen SSONET nutzenden Kommunikationspartnern wird über Sicherheitseigenschaften und zu nutzende –mechanismen ausgehandelt.

## 3.1 Entwickler der Sicherheitsarchitektur

Voraussetzung für die Entwicklung von Sicherheitsarchitekturen ist das Fachwissen zur Problematik von Sicherheitseigenschaften, -mechanismen und deren Integration in Anwendungen. Auf dieser Basis können Komponenten und Schnittstellen sowohl zur Umsetzung als auch Nutzung von Sicherheitsfunktionalität für Anwendungen implementiert werden.

Entwickler von Sicherheitsarchitekturen können mindestens die folgenden unterstützenden Konzepte für andere am Design- und Nutzungsprozeß Beteiligte bereitstellen:

- ein API mit Sicherheitsfunktionalität (z.B. Kryptobibliotheken) für den Anwendungsentwickler,
- eine Nutzerschnittstelle für Systemadministratoren bzw. Endbenutzer (Security Management Interface, SMI),
- ein Rahmenwerk zur Auswahl von Sicherheitsmechanismen nach Kriterien (wie etwa Performance, Kosten, Sicherheit) für Anwendungsentwickler, Systemadministratoren und Endbenutzer,
- Experteninformation als Grundlage für die Bewertung von Sicherheitsmechanismen für Anwendungsentwickler, Systemadministratoren und Endbenutzer,
- Schnittstellen zu anderen (Sicherheits-)Diensten, z.B. Gateways und Verzeichnisdiensten,
- Vorgehensweisen und die notwendige Infrastruktur zur Einbindung neuer Sicherheitsmechanismen und Expertenbewertungen (inkl. von Aktualisierungen) für Anwendungsentwickler, Systemadministratoren und Endbenutzer.

## 3.2 Anwendungsentwickler

Anwendungsentwickler nutzen das von der Architektur bereitgestellte API zur Integration von Sicherheitsmechanismen (Kryptobibliotheken) auf abstrakter Ebene (siehe zum Beispiel [FMRS_94]). Dabei hat sich gezeigt, daß die Abstraktion vom Detail ein gutes Konzept ist: Wenn die Architektur die entsprechende Unterstützung bietet, braucht der Entwickler sich nicht um Details einzelner Mechanismen zu kümmern, sondern kann diese durch abstrakte Methodenaufrufe wie *crypt()* und *sign()* integrieren (siehe z.B. [BaBl_96] oder [PSWW1_98]). Als eine höhere Abstraktionsstufe werden sogenannte Aktions- bzw. Schutzklassen, die die einfache Integration bereits kombinierter Schutzziele für bestimmte Anwendungsfälle ermöglichen, in [Wolf_98] diskutiert, für die wiederum Standardvorgaben gemacht werden sollten. Diese können als eine Bibliothek mit Beispielen für Anwendungs- und Ausnahmefälle für Sicherheitseigenschaften vorliegen.

Anwendungsentwickler haben nicht nur eine Sicht auf die zeitliche Abfolge der einzelnen Aktionen, sondern auch auf die Semantik der Aktionen im Anwendungszusammenhang und können deshalb die Gruppierung von Aktionen mit voraussichtlich gleichen Sicherheitsanforderungen vornehmen. Sie können sowohl Sicherheitsinteressen der einzelnen in einer Anwendung vorkommenden Rollen (über die vor der Kommunikation ausgehandelt wird) vorkonfigurieren als auch besonders sicherheitskritische bzw. –unkritische Aktionen identifizieren und dementsprechend eine Spezifikation sinnvoller oder weniger sinnvoller

Sicherheitsanforderungen (z.B. für Aktion XY keine aufwendigen Anonymisierungsverfahren einsetzen) vornehmen.

Anwendungsentwickler sollten Werkzeuge zur Verfügung haben, um beispielsweise aus dem abstrakten, generischen SSONET-SMI eine konkret auf die Anwendung zugeschnittene Schnittstelle (entweder an das SSONET-SMI angelehnt oder komplett neu) zu bauen. Hierzu können z.B. abstrakte Beispiele für Ausnahmefälle (vgl. [Wolf_98]: einerseits allgemeingültige, wiederverwendbare attributierte Aktionsklassen, andererseits spezifische, anwendungsabhängige Ausnahmefälle) am konkreten Beispiel behandelt werden.

Anwendungsentwickler können wahrscheinlich nur über notwendige Schutzziele, aber nicht über Detaileinstellungen von Sicherheitsmechanismen Vorschriften machen. Aufgrund der Abstraktion sollten sie weder mit Details der Mechanismen (wie Schlüssellängen und Rundenzahlen), noch – im Sinne von flexiblen Sicherheitsarchitekturen – mit einem konkreten Mechanismus (z.B. RSA, DES) konfrontiert werden, also den Mechanismus nur von seiner äußeren Schnittstelle her sehen, um ihn zur Erreichung von Schutzzielen zu integrieren.

Problematisch für Anwendungsentwickler (und indirekt auch für Endbenutzer) ist die Prüfung zugesicherter Eigenschaften eines Programms. Für den Anwendungsentwickler ist die Problematik vor allem dann von Bedeutung, wenn er fremde Bibliotheken in seine Anwendung einbinden will. Ein Ansatz (neben den klassischen Verifikationsansätzen) zur Prüfung von Eigenschaften ist z.B. Programmcode, der einen formalen Beweis seiner Eigenschaften mit sich führt, sog. Proof Carrying Code, siehe [NeLe_97]. So kann der Anwendungsentwickler (und später zur Laufzeit auch der Endbenutzer) die Korrektheit des Programmcodes mit Hilfe eines Proofcheckers prüfen oder von einer Instanz seines Vertrauens prüfen lassen.

## 3.3 Systemadministratoren

Systemadministratoren definieren Sicherheitsanforderungen und konfigurieren Schutzziele und Sicherheitsmechanismen. Im Rahmen mehrseitiger Sicherheit ist das entweder in verschiedenen Abteilungen eines Unternehmens, die verschiedene Sicherheitspolitiken durchzusetzen haben, oder in verschiedenen Unternehmen, aus denen Einzelpersonen miteinander kommunizieren, denkbar. Beide Kommunikationspartner haben auf diese Weise Einstellungen gemäß ihrer Sicherheitspolitik vorliegen. Fraglich ist, inwiefern es überhaupt möglich ist, daß einer der beiden nachgibt, ohne seine Sicherheitspolitik zu verletzen.

Zur Konfigurierung nutzt der Systemadministrator das bereitgestellte SMI (Security Management Interface) und trifft Festlegungen für eine bestimmte Netzumgebung bzw. einen Organisationsbereich oder eine Anwendungsumgebung.

Die Vorkonfigurierung für mehrseitige Sicherheit durch Systemadministratoren kann folgende Ziele haben:

a1) Effizienzsteigerung der Aushandlung (solange der Endbenutzer Einstellungen des Systemadministrators korrigieren kann). Wenn Beteiligte Vorschläge vom gleichen Systemadministrator bekommen, kann ihre Aushandlung evtl. schneller zum Ergebnis kommen.

a2) Durchsetzung einer „mandatory policy“. Der Systemadministrator trifft Einstellungen, die vom Endbenutzer nicht mehr überschrieben werden können.

Systemadministratoren können aber auch in der Rolle des

b) persönlichen Sicherheitsberaters von Endbenutzern gesehen werden. In diesem Falle handelt er ausschließlich im Interesse des Endbenutzers. Dabei auftretende Sicherheitsanforderungen werden im folgenden (Kapitel 3.4) erläutert.

Ein Systemadministrator, der eine „mandatory policy“ (also Mußvorschriften) durchsetzen will, sollte zuerst Regeln zu Zugriffsrechten auf Daten, dann über Daten, die versendet werden dürfen, aufstellen. Erst auf der Basis dieser Regeln (und dem somit eingeschränkten Datenraum) ist es sinnvoll, Regeln zu den die Kommunikation sichernden Mechanismen festzulegen. In Sicherheitsarchitekturen die – wie SSONET – weder Zugriffsrechte noch das Aufstellen von Regeln über versendbare und nichtversendbare Daten unterstützen, sollten demnach über die zu verwendenden Mechanismen keine Mußvorschriften aufgestellt werden. Der Systemadministrator sollte sich also auf die Funktionen, die keine Mußvorschriften aufstellen – a1) und b) – beschränken.

## 3.4 Endbenutzer

Der Endbenutzer nutzt das von der Sicherheitsarchitektur bereitgestellte und ggf. von Anwendungsentwicklern oder Systemadministratoren mofifizierte SMI und kann bei der Formulierung seiner Sicherheitsinteressen Voreinstellungen des Anwendungsentwicklers und Systemadministrators überschreiben. Der Endbenutzer soll auf Wunsch sehen können, welche Entscheidungen (zu Empfehlungen oder Vorschriften bzw. Einschränkungen des SMI) jeweils Entwickler der Sicherheitsarchitektur, Anwendungsentwickler oder Systemadministrator gefällt haben. Dadurch weiß der Endbenutzer, wem er vertraut (vertrauen muß), wenn er diese Einstellungen akzeptiert.

Durch die Abstraktion von Sicherheitsmechanismen auf Schutzziele und ggf. auf umgangssprachlich beschriebene Anwendungsfällen von Sicherheitseigenschaften [siehe Wolf_98] ist es für den Endbenutzer nicht notwendig, Sicherheitsmechanismen oder sogar deren Details zu konfigurieren, solange er die Einstellungen durch vorangegangene Akteure (Anwendungsentwickler, Systemadministrator) akzeptiert; diesen also vertraut. Zur eindeutigen und klaren Trennung der Sicherheitsfunktionalität für Laiennutzer und Sicherheitsexperten ist die Einführung von „advanced“ und „simple“ Nutzermodi anzuraten.

Auch für die Aushandlung sind die bisher genannten Konzepte selektiv anwendbar, wie zum Beispiel Abstraktion, Information und Fehlermeldungen. Bei der Aushandlung kommt der möglichst automatischen Abwicklung besondere Priorität zu. Dies ist notwendig, damit der Nutzer beim Verbindungsaufbau möglichst wenige Teilschritte (Rekonfigurationen) vorzunehmen hat und seine Anwendung trotz Sicherheitsfunktionalität und Aushandlung weitestgehend „reibungslos“ benutzen kann.

In einigen verteilten Anwendungen sind bereits Ansätze oder auch umfassendere Konzepte für die Konfigurierung von Schutzzielen und Sicherheitsmechanismen durch Endbenutzer vorhanden. Die Realisierungen einzelner Teilaspekte der beschriebenen Konzepte finden sich inzwischen u.a. in Programmen wie Pretty Good Privacy, dem Netscape Navigator oder dem Microsoft Explorer.


**Bild 3.** a) Wahl des Verschlüsselungsalgorithmus bei PGP; b) Interaktiver „Konfigurationsdialog“ bei Netscape

Bild 3a) zeigt die Auswahlmöglichkeit eines Verschlüsselungsalgorithmus' in PGP ohne weitere Nutzerinformation. Der Netscape Navigator bietet in seinen neueren Versionen bereits gute Nutzerinformationen. Bild 3b) zeigt einen Warnhinweis über unverschlüsselte Daten.

Die relativ umfassende Lösungsidee des SSONET-Projekts für mehrseitige Sicherheit in verteilten Anwendungen wird u.a. in [PSWW2_98] präsentiert. Ein weiteres Beispiel für die Umsetzung mehrseitiger Sicherheit im Bereich der Telefonie ist in [GaPS_98] beschrieben und wurde in einer Simulationstudie getestet.

## Beratungsdienstleistungen:

Die komplexen Eigenschaften von Schutzzielen und Sicherheitsmechanismen sowie deren Wechselwirkungen mit der Umwelt sind schwer zu überschauen. Deshalb sind Endbenutzer oft nicht mehr in der Lage, ohne entsprechende Beratungsdienstleistungen qualifizierte Entscheidungen zu treffen. Im folgenden Abschnitt wird der spezielle Fall von Beratungsleistungen zur Nutzungszeit von Sicherheitsarchitekturen diskutiert; also nicht die Beratung zur Auswahl einer geeigneten Sicherheitsarchitektur. Im begrenzten Umfang wird diese Beratung durch Anwendungsentwickler und Systemadministratoren geleistet. Darüber hinaus kann der Endbenutzer selbst entscheiden, ob er einen weiteren Berater hinzuziehen will. Beratungsdienstleistung unterteilt sich folgendermaßen:

a) Laden von Konfigurationen/Daten/Informationen von Servern

b) Kommunikation mit Person (entweder vor Ort oder durch entfernte Kommunikation)

Bei Beratung vor Ort sind keine zusätzlichen Sicherheitsmechanismen für Kommunikation notwendig. Bei entfernten Beratungsdienstleistungen erhalten wir eine neue Rolle: Systemunterstützung für entfernte Beratungsdienstleistungen durch den Berater des Endbenutzers. In diesem Fall sind zusätzliche Sicherungsmaßnahmen notwendig. Es ist ein vertrauenswürdiger (oder überprüfbarer; kontrollierbarer) Pfad vom Endbenutzer zum Berater notwendig (damit Endbenutzer nicht heimlich von der NSA beraten wird). Man könnte eine vorkonfigurierte Anwendung „Beratungsdienstleistung“ haben (in der die SSONET-Konzepte angewendet werden), um die Rekursivität abzufangen.

Der Endbenutzer soll sich eine Person seines Vertrauens als Berater wählen können, wenn er dem Systemadministrator nicht vertraut.

Neues Anwendungsbeispiel: Endbenutzer wählt Berater seines Vertrauens und ein Werkzeug zur Systemunterstützung für Remote Consulting. Schickt er ein Duplikat seiner Nutzerschnittstelle an den Berater, sind folgende Sicherheitseigenschaften wünschenswert: verschlüsselt, authentisiert und nicht anonym, Abrechenbarkeit.

Begründungen:

- da keine Realzeitanforderungen bestehen, ist Performance der Sicherheitsmechanismen nebensächlich und sichere Mechanismen sind anwendbar. Eventuell ist keine Aushandlung über Mechanismen notwendig, wenn von Basismechanismen ausgegangen wird, die zunächst bei den Beteiligten vorhanden sind und eingesetzt werden.
- keine Anonymität, da der Berater eh wissen muß, welche Architektur ich benutze; die Identität ist per se nicht schutzbedürftig.
- Zurechenbarkeit, Haftungsfragen: Unterschreibt der Berater alle Ratschläge? Nein, Zurechenbarkeit nur als Ausnahme umsetzen. Für die psychologische Situation der Beratung wird symmetrische Authentikation ausreichen. Müßte der Berater alles signieren, wird er bemüht sein, alle eventuellen Voraussetzungen zur Anwendbarkeit seines Ratschlages aufzuzeigen, was dem Endbenutzer am Ende nicht viel mehr bringt, als wenn er ein Expertensystem benutzen würde, Beispiel: Unter den Umständen v, w, x, y und z ist der Mechanismus a anwendbar.

Schutzwürdige Daten des Kunden (z.B. daß er gerade mit den anonymen Alkoholikern kommunizieren will) sollen nicht an den Berater gelangen, sondern nur das sicherheitstechnische Problem (d.h. Adreßinhalte, Paßwörter, etc. sind auf Anzeige des Beraters auszublenden); es sei denn, das Beratungsgespräch verläuft anonym (was eher unwahrscheinlich ist).

## 3.5 Transparenz zwischen Rollen

Insgesamt gilt, daß sowohl Entwickler der Sicherheitsarchitektur, Anwendungsentwickler als auch Systemadministratoren die Sicherheitsfunktionalität für mehrseitige Sicherheit nicht so stark einschränken (bzw. verdecken) können dürfen, daß der Endbenutzer die Einstellungen nicht wieder rückgängig machen kann. Sie sollen den Endbenutzer nicht bevormunden können, ohne daß dieser es merkt.

Dies bedeutet insbesondere, daß es einen vertrauenswürdigen Pfad vom Entwickler der Sicherheitsarchitektur bis hin zum Endbenutzer geben muß.

# 4 Ausblick

Während die Anforderungen an und Möglichkeiten von Sicherheitsarchitekturen zur Unterstützung und Durchsetzung mehrseitiger Sicherheit inzwischen halbwegs klar sind, so bleibt bzgl. der Umsetzung in kommerzielle Sicherheitsarchitekturen noch viel zu tun. Insbesondere ist es notwendig, gegenseitig voneinander zu lernen, d.h. die Entwickler von Sicherheitsarchitekturen lernen iterativ von den Reaktionen der Benutzer. Diese benötigen zumindest Prototypen von Sicherheitsarchitekturen, um mit dem Themengebiet vertraut zu werden und Wünsche äußern zu können. Dieser Prozeß läßt sich nicht beliebig beschleunigen – umso mehr sollte er rechtzeitig begonnen werden.

## Literatur


BaBl_96 T. Baldin, G. Bleumer: CryptoManager++ – An object oriented software library for cryptographic mechanisms. In: Information Systems Security, Proceedings of the IFIP SEC'96 Conference, Chapman & Hall, London, 1996, 489-491

BKMM_96 I. Biehl, H. Kenn, B. Meyer, B. Müller, J. Schwarz, C. Thiel: LiSA - Eine C++ Bibliothek für kryptographische Verfahren. Digitale Signaturen, DuD Fachbeiträge, Vieweg 1996, 237-248

CDSA_96 Common Data Security Architecture Specification, Release 1.0, October 1996

Cryp Cryptix Library. http://www.systemics.com/software/cryptix-java

Cust_95 H. Custer. Inside Windows NT. Microsoft Press, Redmond, 1995

DuNI_95 F.Dupuy, G.Nilsson, Y.Inoue: The TINA Consortium: Toward Networking Telecommunications Information Services. IEEE Communications Magazine, November 1995, 78-83.

ElNa_87 Clarence A. Ellis, Najah Naffah: Design of Office Information Systems. Springer Verlag, 1987

FMRS_94 Walter Fumy, Gisela Meister, Manfred Reitenspieß, Wolfgang Schäfer (Hrsg.): Sicherheitsschnittstellen – Konzepte, Anwendungen und Einsatzbeispiele. Proceedings des Workshops Security Application Programming Interfaces'94, München, 17./18.November 1994.

GaPS_98 Gunther Gattung, Ulrich Pordesch, Michael J. Schneider: Der mobile persönliche Sicherheitsmanager. GMD Report 24, Juni 1998

GGKL_89 M. Gasser, A. Goldstein, C. Kaufman, B. Lampson: The Digital Distributed System Security Architecture. Proc. 12th National Computer Security Conference 1989, 305-319

HäKK_93 H. Härtig, O. Kowalski, W. E. Kühnhauser: The BirliX Security Architecture. Journal of Computer Security 2/1 (1993) 5-21

Kran_96 A. Krannig: PLASMA Platform for Secure Multimedia Applications. 2nd IFIP Communications and Multimedia Security, Chapman & Hall, London 1996, 1-12

Linn_90 J. Linn: Generic Security Service Application Program Interface. 2nd USENIX Security Symposium, 1990, 33-53

MPSC_93 S. Muftic, A. Patel, P. Sanders, R. Colon, J. Heijnsijk , U. Pulkkinen: Security Architecture For Open Distributed Systems. John Wiley & Sons Ltd. Baffins Lane, Chichester West Sussex PO19 1UD, U.K.

NeLe_97 G. C. Necula, P. Lee: Research on Proof-Carrying Code for Untrusted-Code Security; In: Proceedings of the 1997 IEEE Symposium on Security and Privacy, Oakland, 1997.

NoHe_90 Claus Noack, Dietmar Hennig: Systemsicherheit unter Unix. Carl Hanser Verlag München Wien, 1990.

OMG_95 OMG. CORBA Security. Document Number 95-12-1. December 1995

PGP Pretty Good Privacy. http://www.pgp.com

PSWW1_98 A. Pfitzmann, A. Schill, A. Westfeld, G. Wicke, G. Wolf, J. Zöllner: A Java-based distributed platform for multilateral security. In: Winfried Lamersdorf, Michael Merz: Trends in Distributed Systems for Electronic Commerce. Proceedings of TREC ‘98 Hamburg, Germany: June 3.-5. 1998, LNCS 1402

PSWW2_98 A. Pfitzmann, A. Schill, A. Westfeld, G. Wicke, G. Wolf, J. Zöllner: Flexible mehrseitige Sicherheit für verteilte Anwendungen. Angenommen für KiVS ’99.

REMO_93 REMO: Referenzmodell für sichere IT-Systeme - Überblick. EISS, GMD, IABG, Siemens, TELES 1993

RSA RSAREF. http://www.rsa.com/rsalabs/newfaq/q174.html

SBHS_97 S. Staamann, L. Buttyán, J-P. Hubaux, A. Schiper, U. Wilhelm: Security in the Telecommunications Information Networking Architecture - the CrySTINA Approach. TINA'97 Conference, Santiago, Chile, November 17-20, 1997, proceedings published by IEEE Publications 1997.

Schi_97 A. Schill. DCE - Das OSF Distributed Computing Environment: Grundlagen und Anwendung (2., erweiterte Auflage); Springer-Verlag, 1997

Secu SECUDE. http://www.darmstad.gmd.de/secude

SET_97 Mastercard Inc., Visa Inc.: Secure Electronic Transactions (SET) Version 1.0 - Book 1: Business Descriptions, Book 2: Programmer's Guide, Book 3: Formal Protocol Specification. Report, May 31, 1997

SSONET http://wwwrn.inf.tu-dresden.de/RESEARCH/ssonet/ssonet.html

SSL SSLeay. http://www.psy.uq.edu.au:8080/~ftp/Crypto/

StNS_88 J. G. Steiner, Clifford Neuman, Jeffrey I. Schiller: Kerberos: An Authentication Service for Open Network Systems. USENIX Conference Proceedings, Feb. 1988, 191-202

Waid_96 M. Waidner: Development of a Secure Electronic Marketplace for Europe. ESORICS '96 (4th European Symposium on Research in Computer Security), Rome, LNCS 1146, Springer-Verlag, Berlin 1996, 1-14

Wiew_96 E. Wiewall. Secure Your Applications with the Microsoft CryptoAPI. In Microsoft Developer Network News, 3/4 1996, Microsoft Press

Wolf_98 Gritta Wolf: Generische, attributierte Aktionsklassen für mehrseitig sichere, verteilte Anwendungen. Proceedings des Workshops „Sicherheit und Electronic Commerce“, Essen, 1./2. Oktober 1998