Veröffentlicht in: R. Steinmetz (Hrsg.): Kommunikation in Verteilten Systemen (KiVS), 11. ITG/GI-Fachtagung Darmstadt, 2.-5. März 1999, Springer, 130-143

Flexible mehrseitige Sicherheit für verteilte Anwendungen[♣]

A. Pfitzmann**, A. Schill*, A. Westfeld**, G. Wicke**, G. Wolf*, J. Zöllner*

Technische Universität Dresden, 01062 Dresden
*Institut für Betriebssysteme, Datenbanken und Rechnernetze
**Institut für Theoretische Informatik
{pfitza, schill, westfeld, wicke, g.wolf, zoellner}@inf.tu-dresden.de

Zusammenfassung. Wir stellen eine prototypische Implementierung einer Sicherheitsarchitektur vor. Sie unterstützt die Nutzer und Entwickler verteilter Anwendungen bei der Umsetzung bzw. Integration von mehrseitiger Sicherheit. Schutzziele wie Vertraulichkeit und Integrität und ihnen zugeordnete kryptographische Mechanismen werden sowohl nutzer- als auch anwendungsbezogen formulierbar bzw. konfigurierbar. Der konkrete Schutz einer Kommunikation, z.B. über offene Datennetze, wird zwischen den Partnern ausgehandelt. Heterogenen Anforderungen der Nutzer bzw. Applikationen und heterogenen Eigenschaften der Schutzmechanismen wird durch Architekturkomponenten für Konfigurierung (mit Modulen für Rating und Performance-Test) und Aushandlung sowie Sicherheitsgateways Rechnung getragen. Die Architektur setzt jeweils lokal sichere Basissysteme voraus und ermöglicht darauf aufbauend flexible mehrseitige Sicherheit für verteilte Anwendungen.

1 Einführung

Der im folgenden vorgestellten Sicherheitsarchitektur liegt das Prinzip der mehrseitigen Sicherheit [vgl. FePf_97] zugrunde. Es besagt, daß alle Nutzer eines informationstechnischen (Teil-)Systems in der Wahrung ihrer Schutzinteressen gleichberechtigt zu unterstützen sind, und zwar so, daß ihre Sicherheit so wenig wie möglich von der Gutwilligkeit anderer abhängt. Besonderer Wert wird auf die Möglichkeit differenzierter Widerspiegelung der Schutzinteressen gelegt.

Den Nutzern soll ein Mittel in die Hand gegeben werden, das ihnen erlaubt, ihr Schutzniveau bei kommunizierenden Applikationen weitgehend selbst zu bestimmen, mit dem Kommunikationspartner darüber zu verhandeln, eventuelle Konflikte aufzudecken und zu lösen. Die Architektur unterstützt den Nutzer u.a. durch Abstraktion und Automatisierung von Abläufen. Eine prototypische Implementierung

♣ Diese Arbeit wurde finanziell unterstützt vom Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF).

ist im Rahmen des BMBF-Projektes SSONET (Sicherheit und Schutz in offenen Datennetzen) entstanden [PSWW_98].

Ziel des vorliegenden Beitrags ist es, ein im Gegensatz zu anderen existierenden Ansätzen flexibleres Konzept einer Sicherheitsplattform vorzustellen, das explizit am Prinzip der mehrseitigen Sicherheit orientiert ist und dieses durch neue Abstraktions-, Verhandlungs- und Konfigurierungsmechanismen praktisch umsetzt.

Mittlerweile stehen zahlreiche Sicherheitsmechanismen und –plattformen mit unterschiedlichen Zielsetzungen zur Verfügung, die in bezug auf Kryptoverfahren und Protokolle recht weit entwickelt sind. Als Beispiele seien insbesondere die CORBA Security Services [OMG_97], die Mechanismen im Bereich der Internet Security wie Secure Socket Layer [SSL_97] und die Weiterentwicklung als Transport Layer Security [TLS_97]) sowie ausgewählte Sicherheitsplattformen wie *SECUDE* - Security Development Environment [SECU_98], *LiSA* - Library for Secure Applications [BKMS_96], CryptoManager++ [Kann_94] und PLASMA - Platform for Secure Multimedia Applications [GeKo_95, Kran_96] genannt.

Im Gegensatz zum hier verfolgten Ansatz realisieren sie in der Regel jedoch fest vorgegebene, einheitliche Sicherheitskonzepte. Sie unterstützen kaum heterogene Szenarien (wie Electronic Commerce) mit unterschiedlichen Sicherheitspolitiken im Sinne mehrseitiger Sicherheit, wechselnden Anforderungen an Sicherheitsmechanismen und unterschiedlicher Leistungsfähigkeit der Endsysteme durch eine nutzer- und anwendungsspezifische Aushandlung und Konfigurierung.

Unser Beitrag gliedert sich wie folgt: In Kapitel 2 wird die Gesamtarchitektur des SSONET-Systems vorgestellt und insbesondere erläutert, wie die Aushandlungs- und Konfigurierungsmechanismen realisiert wurden und ablaufen. Weiterhin wird das Konzept der Sicherheitsgateways als Mechanismus zur Transformation sicherer Kommunikationsvorgänge bei inkompatiblen Systemvoraussetzungen oder inkompatiblen Einschätzungen der Sicherheit kryptographischer Mechanismen beschrieben. Kapitel 3 detailliert die Realisierung und Validierung der Ergebnisse und Kapitel 4 gibt einen Ausblick auf weiterführende Fragestellungen.

2 Die SSONET-Architektur für flexible mehrseitige Sicherheit

2.1 Übersicht

Die SSONET-Architektur liegt als prototypische Implementierung in JAVA™ vor und ist für den Einsatz auf heterogenen Plattformen geeignet. Als generische Systemarchitektur für mehrseitige Sicherheit hat sie folgende konstituierende Architekturbausteine: eine Konfigurations- und eine Aushandlungskomponente sowie ein umfangreiches sog. Security Management Interface und ein Application Programming Interface. Über dieses können Anwendungen auf Sicherheitsdienste und -mechanismen in Bibliotheken zugreifen (Abbildung 1).

Das Security Management Interface (SMI, vgl. Abschnitt 2.2.1) bietet dem Nutzer Oberflächen zur Eingabe und Modifikation seiner Sicherheitsinteressen an, dazu dienen sowohl Grund- als auch Anwendungskonfiguration.

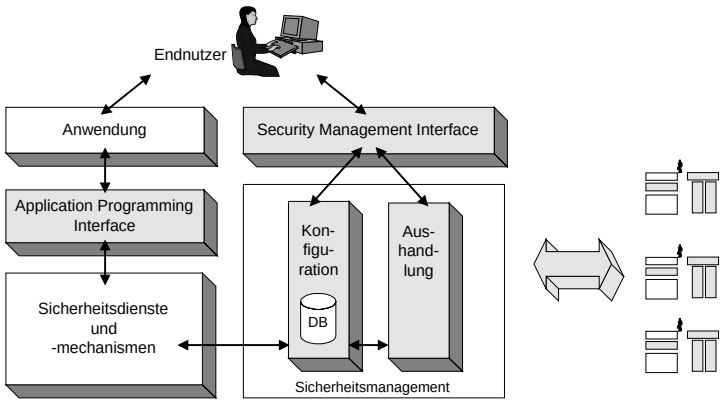

Abbildung 1: Die SSONET-Architektur

Die zuvor vom Anwendungsentwickler festgelegten Anwendungsanforderungen stellen das Mindestmaß an notwendiger Sicherheitsfunktionalität für eine Anwendung bzw. deren Aktionen dar. Ist das lokale System und die Anwendung konfiguriert und will der Endbenutzer die Anwendung starten, so wird eine möglichst automatisch ablaufende Aushandlungsphase (siehe Abschnitt 2.3) angestoßen, im Laufe derer sich die Kommunikationspartner auf eine Kommunikationsgrundlage einigen. Anschließend nutzt die Anwendung entsprechend der ausgehandelten Konfiguration die über das API zugreifbaren Mechanismen (beispielsweise TripleDES im CBC-Modus für Vertraulichkeit) für die zu schützende Kommunikation.

2.2 Lokales Sicherheitsmanagement

Bei der Nutzung der Sicherheitsplattform muß der Anwender beim Umgang mit heterogenen und für verschiedene Systeme und Anwendungen unterschiedlich geeigneten Sicherheitsmechanismen unterstützt werden. Die Plattform muß also von spezifischen Details der Sicherheitsmechanismen, der Implementierung oder der Mechanismenintegration abstrahieren, um dem Nutzer eine homogene Schnittstelle anzubieten.

2.2.1 Konfigurierung

Die SSONET-Architektur beinhaltet im Grundzustand ein Standardeinstellungs-Set für die Grundkonfiguration. Diese berechnet sich teilweise aus den im Rating (siehe Abschnitt 2.2.2) gewonnenen Erkenntnissen über Sicherheit, Performance und Kosten für den Einsatz der Mechanismen und bezieht sich auf die standardmäßig mit der Architektur mitgelieferten Mechanismen. Der Nutzer kann alternativ aber auch ein Rating erstellen lassen, das die Sicherheitsmechanismen entsprechend seiner Wichtung der Kriterien (Sicherheit, Performance, Kosten) bewertet und so in eine

halbautomatisch erstellte Präferenzliste einordnet. Beide Listen – das Standard-Set als auch die benutzergewichtete Präferenzliste – können in den Dialogfenstern der Grundkonfiguration über das SMI (Security Management Interface) entsprechend den Nutzerwünschen (weiter) modifiziert werden.

Im linken Teil der Abbildung 2 ist als Beispiel das Fenster für das Editieren der Präferenzliste zum Schutzziel Zurechenbarkeit (Erstellen bzw. Akzeptieren digitaler Signaturen) gezeigt. Im rechten Teil der Abbildung 2 ist wiederum als Beispiel ein Detailfenster für die Auswahl der Betriebsarten für den Sicherheitsmechanismus RSA abgebildet. Weitere konfigurierbare Details sind die zu verwendende Schlüssellänge und der zu verwendende Hash-Mechanismus.

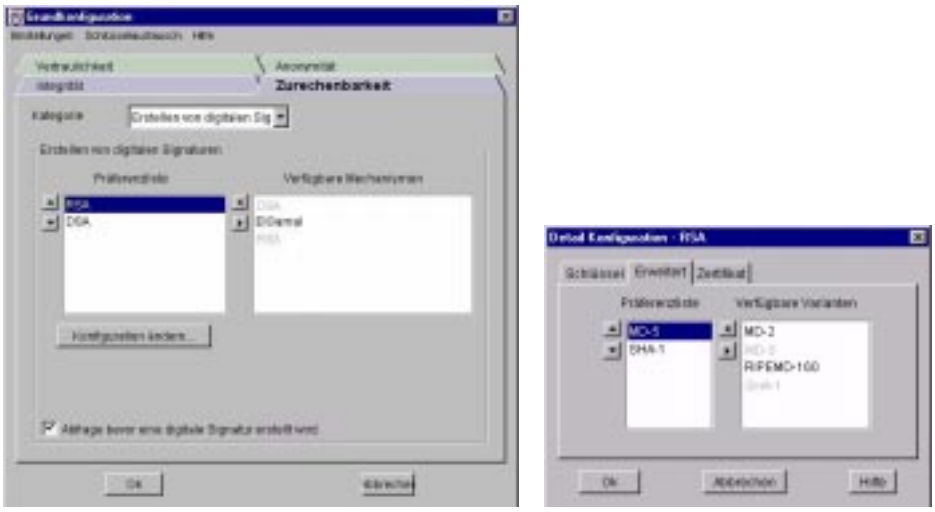

Abbildung 2: Präferenzliste und Detailkonfiguration der Mechanismen

Für auf der SSONET-Plattform lauffähige Anwendungen werden durch den Anwendungsentwickler oder Experten Anwendungsanforderungen festgelegt. Die Anwendungsanforderungen stellen ein aus Sicht der Experten nicht zu unterschreitendes Mindestmaß an Sicherheitsfunktionalität dar, das für eine Anwendung notwendig ist. Die geforderten Schutzziele werden pro Anwendungsaktion festgelegt und können aus den fünf Präferenzen `always` (immer), `if possible` (möglichst), `don't care` (egal), `if needed` (wenn nötig) und `never` (nie) ausgewählt werden [vgl. auch GaPS_98].

Die Präferenzen dienen direkt der Beeinflussung des Aushandlungsprozesses über die Schutzziele in Phase 1 (siehe Abschnitt 2.3.1). Dabei sind `always` und `never` die härtesten Forderungen und bedeuten, daß der Nutzer generell nicht bereit ist, von seinen Sicherheitsanforderungen abzuweichen. Für ein Schutzziel `never` als harte Forderung einzustellen, kann zum Beispiel sinnvoll sein für Zurechenbarkeit, wenn anonym kommuniziert werden soll. Die Anwendungsanforderung kann `don't care` (egal) sein, wenn während einer Aktion keine sensiblen bzw. besonders

schutzbedürftigen Daten übertragen werden. Die Präferenz `if possible` wird benötigt, um zu kennzeichnen, daß die Erreichung eines Schutzzieles sinnvoll ist, aber nicht zwingend notwendig. Analoges gilt für die Einstellung `if needed`; es ist in manchen Fällen (z. B. wenn nicht anonym kommuniziert werden soll) nicht zwingend notwendig, „keine Zurechenbarkeit" hart zu fordern, der Nutzer könnte aber durchaus auch bereit sein, Zurechenbarkeitsmechanismen anzuwenden. Die Tabelle 1 gibt beispielhaft die festgelegten Anwendungsanforderungen für die Aktion „Abschicken der Bestellung" an.

	Partner A (sender)	Partner B (recipient)
	Customer	Merchant
SecurityGoal \ Action	SendOrder	ReceiveOrder
confidentiality	if needed	always
anonymity	never	never
accountability	if needed	always
integrity	always	always

Tabelle 1: Beispiel für Anwendungsanforderungen für eine Aktion

Startet der Endbenutzer eine Anwendung, so werden im Dialogfenster für die Anwendungskonfiguration die zuvor festgelegten Anwendungsanforderungen pro Aktion angezeigt. Nutzer können entsprechend ihrer Schutzziele andere Einstellungen wählen. Dabei gelten die gleichen 5 Präferenzen wie bei der Festlegung der Anwendungsanforderungen. Zur zusätzlichen Charakterisierung der Präferenzen `always` und `never` wurde die Auswahl-Box `negotiable` (verhandelbar) eingeführt, um in der späteren Aushandlung interaktive, partnerbezogene Entscheidungen über Schutzziele zu ermöglichen.

Während der Einstellung der Anwendungskonfiguration werden entsprechend der ausgewählten Präferenz „dynamisch" Icons angezeigt, die insbesondere Nicht-Experten unterstützen und ihnen die Benutzung der SSONET-Plattform vereinfachen sollen. Zum Beispiel wird für die Forderung des Schutzzieles Vertraulichkeit ein verschlossener Briefumschlag als Symbol der geschützten Nachrichteninhalte angezeigt, andernfalls eine Postkarte als Symbol für „ungehindertes Mitlesen auf der Übertragungsstrecke".

Plausibilitätstest: Bevor Grund- und Anwendungskonfiguration in einer sicheren Datenbank abgespeichert werden, testet die Sicherheitsplattform mit einem Plausibilitätstest die Konsistenz der Einstellungen. Für diesen Test werden Regeln verwendet, die z.B. folgenden Klassen angehören können: gegenseitiger Ausschluß von Schutzzielen, gemeinsame Anwendung zweier Schutzziele oder zwingende Auswahl eines Algorithmus aufgrund von Schutzzielkombinationen. Ausgewählte Beispielregeln sind in Tabelle 2 beschrieben. Die Erfüllung einiger solcher Regeln kann bereits durch ein geeignetes Oberflächen- bzw. Systemdesign erreicht werden (harte Regeln); bei anderen Regeln mit Empfehlungscharakter bedarf es der Fehlermeldung an den Nutzer und einer Nutzerentscheidung über das weitere Vorgehen zur Beseitigung inkonsistenter Einstellungen.

wenn	dann: durch System automatisieren	dann: Nutzerinformation
Sender-anonymität	Zurechenbarkeit nicht wählbar → keine personenbezogenen digitalen Signaturen leistbar → Verschlüsselung automatisch ausgewählt	• Hinweis, daß Inhaltsdaten keinen Absenderbezug enthalten sollten; • Hinweis, daß Verschlüsselung empfehlenswert ist und deshalb ausgewählt wurde
Empfänger-anonymität	Zurechenbarkeit nicht wählbar → keine Empfangsquittungen leistbar → Verschlüsselung automatisch ausgewählt	• Hinweis, daß Verschlüsselung empfehlenswert ist und deshalb ausgewählt wurde
Unbeobacht-barkeit der Kommuni-kations-beziehung	Verkettbarkeit über Inhalte vermeiden → Verschlüsselung (indeterministisch) automatisch augewählt zeitliche Verkettbarkeit vermeiden → dummy traffic (Senden bedeutungsloser Nachrichten)	• Hinweis, daß Verschlüsselung empfehlenswert ist und deshalb ausgewählt wurde • Hinweis, daß Verfahren aufwendig sind und möglicherweise Performance-Einbußen hervorrufen können
Zurechen-barkeit	Integrität automatisch ausgewählt; Verfahren Digitale Signatur wird verwendet	• Hinweis, daß Zurechenbarkeit auch die Beweisbarkeit gegenüber Dritten ermöglicht; Willensäußerung ist notwendig

Tabelle 2: Beispielregeln für den Plausibilitätstest

2.2.2 Bewertung von Mechanismen

Performance-Test: Durch die Ausführung eines Performance-Tests wird die Leistungsfähigkeit der auf dem lokalen System verfügbaren Sicherheitsmechanismen für jedes Schutzziel unter den aktuellen Gegebenheiten (z.B. CPU-Last durch verschiedene Anwendungen) getestet. Die Ergebnisse dieses Tests[1] werden für den Nutzer veranschaulicht. Darüber hinaus fließen die Meßergebnisse automatisch in das Rating (s.u.) ein, wo sie einen Teilwert zur Gesamtbewertung der Sicherheitsmechanismen beisteuern.

Selbst auf dem eigenen lokalen System können in bestimmten Systemsituationen unterschiedliche Grenzen für die Einsetzbarkeit leistungsstärkerer oder -schwächerer Mechanismen gelten. Somit kann es durchaus sinnvoll sein, in regelmäßigen Abständen oder in speziellen Systemsituationen erneut einen Performance-Test durchzuführen, um aufgrund von aktuellen Testergebnissen Entscheidungen über zu nutzende Mechanismen zu treffen.

Rating: Das Rating bildet ein Rahmenwerk zur interaktiven Ermittlung präferierter Sicherheitsmechanismen. Es wurde integriert, um Nutzer und insbesondere Nicht-Experten während der Konfigurierung zu unterstützen. Der Endbenutzer kann sich anzeigen lassen, wie die Mechanismen von Experten bewertet werden und wird in die

[1] Auftretende Unterschiede der Dauer von Ver- bzw. Entschlüsselung setzen sich aus mehreren Komponenten zusammen:
1. Algorithmenspezifika
2. Spezifika des Java-Interpreters bzw. Compilers und der virtuellen Maschine.

Lage versetzt, Wichtungen für einzelne Bewertungskriterien zu vergeben, anhand derer programmintern unter Zuhilfenahme der Expertenbewertung mittels Formel (1) ein Gesamtwert G_{Mech} für jeden Mechanismus errechnet wird:

$$G_{Mech} = \sum_{i=1}^{k} a_i \cdot v_i \qquad (1)$$

Dabei ist *a* die Wichtung durch den Nutzer für ein Kriterium, also der Faktor zur Multiplikation mit den Expertenwerten *v*. Der Index *i* geht über die folgenden Kriterien *k*:

1. Sicherheit des Algorithmus $v_{1,1}$ (kryptografische Stärke, Schlüssellänge) und Sicherheit der Implementierung $v_{1,2}$ (korrekte Umsetzung einer vollständigen Spezifikation), wobei

$$v_1 = min\ (v_{1,1}, v_{1,2}) \qquad (2)$$

2. Performance (durch Messung ermittelt),
3. niedrige Beschaffungskosten und Kosten für Einrichtung und Wartung (Schätzwert).

Weiterhin kann der Nutzer für die Kriterien „minimaler Durchsatz“ (durch Messung ermittelt) und „maximale Kosten“ jeweils den gewünschten unteren bzw. oberen Grenzwert angeben.

Der Vergleich aller verfügbaren Mechanismen auf dieser Basis liefert als Ergebnis ein benutzergewichtetes Rating. Dies stellt die für alle verfügbaren Mechanismen ermittelten Gesamtwerte in einem Balkendiagramm zueinander ins Verhältnis. Je nach Nutzergewichtung (abhängig vom Zweck und den Bedingungen des Einsatzes) kann jedoch der Endbenutzer auch unter bestimmten Voraussetzungen unsichere bzw. gebrochene Mechanismen zur Auswahl zulassen. Die Gesamtbewertung erfolgt pro Schutzziel, da einige Mechanismen für mehrere Schutzziele anwendbar sind.

Durch die gezeigten Mechanismen Konfigurierung, Plausibilitätstest, Performance-Test und Rating wird dem Nutzer ermöglicht, die Heterogenität von Sicherheitsmechanismen einzuschätzen, davon zu abstrahieren und seine Sicherheitsinteressen zu formulieren.

2.3 Aushandlung gesicherter Aktionen

Um mehrseitig sichere Kommunikation zu ermöglichen, müssen sich die Partner auf eine von den Beteiligten akzeptierte gemeinsame Basis (Schutzziele und Mechanismen) einigen. Um heterogene Sicherheitsanforderungen abstimmen zu können, werden Aushandlungsprotokolle benötigt. Die Aushandlung in SSONET geht von folgenden Voraussetzungen aus:

1. Es werden jeweils bilaterale Aktionen verhandelt, d.h. an einer Aushandlung sind nur zwei Partner beteiligt[2].
2. Jeder Teilnehmer hat Einstellungen getroffen (oder die Default-Konfiguration aktiviert), aus denen Aushandlungsvorschläge generiert werden. Sie befinden sich persistent in Dateien zur Grundkonfiguration (Mechanismen) und zur

[2] Eine multilaterale Aushandlung setzt sich aus mehreren bilateralen Aushandlungen und Konfigurationen zusammen.

Anwendungskonfiguration (Schutzziele). Die Schutzziele sind dabei nach vom Anwendungsprogrammierer definierten Aktionen der Anwendung (z.B. Geschäftsbeziehungs-Transaktionen) aufgegliedert.

3. Die Einstellungen (Grund-, Anwendungskonfiguration) für alle potentiellen Partner sind gleich (d.h. unabhängig vom konkreten Partner) und erlauben eine weitgehend automatische Aushandlung.
4. In Abhängigkeit vom Partner können Abweichungen von den Einstellungen in der Anwendungskonfiguration, und zwar bei den Maximalforderungen `always` bzw. `never` zugelassen (Einstellung `negotiable`) und interaktiv entschieden werden. Damit ist eine Berücksichtigung des konkreten Partners (und damit unterschiedliche Aushandlung trotz gleicher Einstellungen für alle potentiellen Partner) möglich.
5. Die Architektur implementiert eine Tendenz „pro Sicherheit“, die bei unentschiedenen Situationen zum Tragen kommt.
6. Mit der Architektur mitgeliefert wird ein Mechanismus für digitale Signaturen zum Schutz der Aushandlung vor Verfälschung. Der jeweils zugehörige nutzergenerierte öffentliche Schlüssel ist zertifiziert[3].

Der Ablauf erfolgt in zwei Phasen. Die erste Phase (Abbildung 3) beinhaltet die Aushandlung der Schutzziele bezüglich einer auszuführenden Aktion der zu schützenden Applikation; eine zweite Phase behandelt die Mechanismen zu deren Umsetzung.

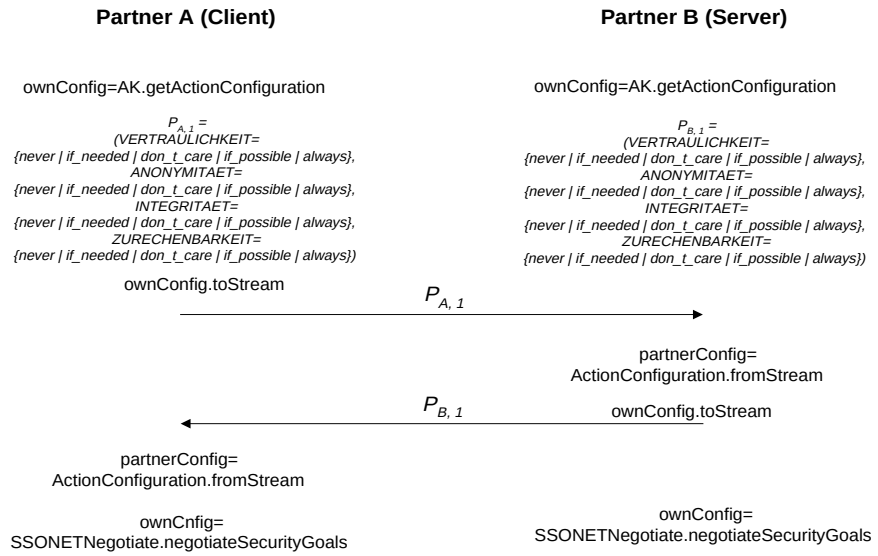

Abbildung 3: Aushandlung der Schutzziele für Aktion

[3] Zertifikate können dezentral erzeugt und auf Pseudonyme ausgestellt werden. Sie müssen nicht unbedingt rechtsverbindlich sein.

Aushandlung der Schutzziele: Mit den Aushandlungsvorschlägen (Abbildung 3) werden zunächst nur die Präferenzen, ohne eine eventuelle partnerbezogene Charakterisierung der Maximalforderungen (`never` und `always`) durch `negotiable`, übertragen. Jeder empfangene Aushandlungsvorschlag s_B wird mit dem eigenen Vorschlag s_A über eine Entscheidungstabelle (Tabelle 3) verknüpft, so daß bis auf zwei Ausnahmen automatisch ein Ergebnis ermittelt werden kann.

$s_A \mid s_B$	never	if needed	don't care	if possible	always
never	nein	nein	nein	nein	k.E.
if needed	nein	nein	nein	ja	ja
don't care	nein	nein	*	ja	ja
if possible	nein	ja	ja	ja	ja
always	k.E.	ja	ja	ja	ja

$s_A \setminus s_B$	von Partner A \ Partner B übermittelte Präferenz		
ja	Schutzziel durchsetzen	*	Einigung auf Standardvorschlag
nein	Schutzziel ignorieren		Tendenz „pro Sicherheit" wirkt
		k.E.	keine Einigung

Tabelle 3: Entscheidungstabelle für Präferenzen

Falls keine Einigung (k.E.) erzielt wird, liegt ein Konflikt vor, der zu einer Interaktion mit eventueller Modifizierung der Maximalforderungen führt. Eine Einigung auf einen Standardvorschlag bedeutet, daß mitgelieferte Vorgaben Dritter wie des Anwendungsprogrammierers oder der Architektur (Tendenz „pro Sicherheit") bzw. die Verfügbarkeit des Mechanismus den Ausschlag geben.

Ein auftretender Schutzziel-Konflikt durch eine strukturierte Liste mit den in Konflikt stehenden Einstellungen visualisiert. Zusätzlich gibt es eine Beschreibung der Folgen einer Modifizierung von verhandelbaren Maximalforderungen in einem separaten Fenster. Es erfolgt jetzt eine Entscheidung aufgrund der Option `negotiable` durch Nachverhandlung mit erneuter Übertragung (endgültiger) Vorschläge. Dazu muß von jedem, der bei den internen Präferenzen `negotiable` gewählt hat, interaktiv eine Entscheidung getroffen werden:

- bei `never,negotiable` auf `never` oder auf `if needed` und
- bei `always,negotiable` auf `always` oder auf `if possible` .

Sollte hier keine Einigung zustande kommen (k.E.), so kann die Aktion der Applikation (bis zur nächsten Änderung der Konfiguration bei einem der beiden Partner und erneuter Aushandlung) nicht ausgeführt werden. In allen anderen Fällen findet Kommunikation statt.

Aushandlung der Mechanismen: Für gemeinsame Schutzziele werden anschließend gemeinsam akzeptierte Schutzmechanismen ausgehandelt. Die Mechanismen sind in einer lokalen Präferenzliste geordnet. Alle Mechanismen sind der Architektur mit Namen bekannt. Die Aushandlung geschieht nun, indem sich beide ihre nach Präferenz geordneten Mechanismen mitteilen und anschließend jeweils lokal eine Schnittmengen- und Maximumbildung vornehmen. Bei Gleichstand mehrerer Mechanismen entscheidet die Präferenz der Architektur („pro Sicherheit"-Strategie).

Ein Mechanismen-Konflikt (der bei leerer Schnittmenge vorliegt) ist lösbar, falls die Sicherung der Kommunikation (d.h. hier der aktuellen Aktion der Anwendung) durch

a) einen Sicherheitsgateway-Dienst,
b) die kombinierte Verwendung von Mechanismen beider Partner oder
c) das Nachladen fehlender Mechanismen

erreicht werden kann und der oder die Partner dies wollen. Sonst kann keine Kommunikation stattfinden. Das bedeutet auch, daß die Anwendung jetzt keine unsichere Kommunikation aufbaut, da dies ein Widerspruch zu dem bereits erfolgreich ausgehandelten Schutzziel wäre.

Beispiel: Für eine Teleshopping-Applikation haben die Kommunikationspartner folgende Präferenzen:

- Kunde A: für A sichere Kommunikation > keine Kommunikation > für A unsichere Kommunikation.
- Händler B: für B sichere Kommunikation > keine Kommunikation > für B unsichere Kommunikation[4].

Ein Konflikt kann nun in der Weise auftreten, daß der Kunde A eine Kombination vorschlägt, die zwar für ihn sicher ist (bzw. die A für sicher hält), aber nicht für den Händler B (bzw. die B nicht für sicher hält). Das kann beispielsweise bei Schutzzielen der Fall sein (der Kunde fordert Vertraulichkeit für die Aktion „Bestellung senden“ während der Händler dies ablehnt) oder bei Mechanismen (der Kunde will obige Aktion mit IDEA im CFB-Modus vertraulich halten, während der Händler nur über DES im ECB-Modus verfügt).

Für das Schutzziel bedeutet das: A bewegt sich mit seinen Entscheidungen im Bereich `if possible` / `always`, B im Bereich `never` / `if needed`. Die besten verbleibenden Lösungen nach Tabelle 3 bei Schutzzielpaarungen (s_A, s_B) für den Kunden A wären (`if possible`, `if needed`)→ja und (`always`, `if needed`)→ja. Schlechter wäre (`always`, `never`)→k.E. Die schlechteste verbleibende Lösung für A wäre (`if possible`, `never`)→nein. Für den Händler B kehrt sich das um, die schlechteste Lösung für den Kunden ist seine beste und umgekehrt. Folglich wurde im ersten Aushandlungsvorschlag die Kombination (`always`, `never`) getroffen, da jeder unabhängig vom anderen konfiguriert, d. h. entscheidet. Für die Nachverhandlung bedeutet das, daß einer von beiden oder beide die Präferenzen ändern müssen, wenn die Anwendung kommunizieren soll.

---

[4] Eine alternative denkbare Reihenfolge wäre: für Händler B sichere Kommunikation > für B unsichere Kommunikation > keine Kommunikation. Sie ist dann interessant, wenn Schutz durch Mechanismen unter bestimmten Rahmenbedingungen, z. B. durch Leistungen aus Versicherungsverträgen kompensiert werden kann, wobei hier Haftungsfragen klärbar sein müssen.

2.4 Sicherheitsgateways

Der notwendige Aushandlungsprozeß soll einerseits so automatisiert wie möglich ablaufen und anderseits alle Möglichkeiten für eine gesicherte Kommunikation entsprechend der Vorgaben der Partner ausschöpfen. Es sind grundsätzlich zwei Möglichkeiten denkbar, warum diese Ziele nicht erreichbar sein können:

- Die Benutzer haben unterschiedliche Präferenzen bezüglich der Schutzziele.
- Die Kommunikationspartner können sich zwar auf zu verwendende Schutzziele einigen, jedoch nicht bezüglich der zu verwendenden Mechanismen.

Im ersten Fall ist manuelles Eingreifen notwendig, da hier Interessenkonflikte vorliegen, die nur durch persönliche Entscheidungen der Nutzer gelöst werden können (siehe Abschnitt 2.3). Im zweiten Fall bestehen keine grundsätzlichen Meinungsverschiedenheiten, aber die technische Umsetzung ist zunächst nicht möglich, läßt sich jedoch durch Sicherheitsgateways erreichen, welche die Transformation von verschiedenen Mechanismen und Verfahren auf nachweisbare und rechtsverbindliche Art und Weise übernehmen und die Verbindung zwischen beiden Parteien herstellen.

Mittels des Gateways wird die von einer oder beiden Seiten als unsicher oder nicht realisierbar empfundene Verbindung (hier zwischen Kunde und Händler) in zwei aufgetrennt, der die jeweils angeschlossenen Teilnehmer (Kunde/Gateway bzw. Gateway/Händler) vertrauen können. Das Gateway nimmt eine nachprüfbare Konvertierung vor, für die es haftet. Entsprechend kann es sich seinen Aufwand vergüten lassen. Die folgende Tabelle 4 faßt zusammen, was Gateways leisten können.

Transformation	Wirkung
Digitale Signatur → Digitale Signatur	Integrität bleibt gewährleistet, ohne daß dem Gateway vertraut werden muß, da jeder Mißbrauch Dritten beweisbar ist
Symmetrische Authentikation → Symmetrische Authentikation	Integrität bleibt nur gewährleistet, wenn das Gateway vertrauenswürdig ist. Betrug ist feststellbar, wenn der Agent nicht zu jedem Zeitpunkt alle Kommunikationswege zwischen den Partnern kontrollieren kann (Beachtung der Systemgrenzen)
Konzelation → Konzelation	Vertraulichkeit bleibt nur gewährleistet, wenn das Gateway vertrauenswürdig ist, da Kenntnisnahme der geheimen Nachricht notwendig; unerlaubte Weiterverbreitung ist nicht feststellbar
Anonymität des Senders	Analog zur Verwendung von Mixen etc.: Gateway muß vertrauenswürdig sein
Anonymität des Empfängers	Realisierbar (z.B. Broadcast-Generierung durch Gateway)
Unbeobachtbarkeit der Kommunikations-beziehung	Realisierbar

Tabelle 4: Transformierbarkeit von Schutzzielen

Werden Kombinationen transformiert (also Elemente der Potenzmenge aus digitaler Signatur, symmetrischer Authentikation und Konzelation), so kombinieren sich die Wirkungen entsprechend, es gibt keine Wechselwirkungen.

3 Prototypische Implementierung

Dem Anwendungsentwickler stehen mehrere Möglichkeiten zur Verfügung, seine Anwendung kommunizieren zu lassen. In JAVA™ kann dies z.B. über RMI (Remote Method Invocation) oder über Socket-Verbindungen geschehen. Letzteres wurde für die SSONET-Architektur gewählt. Es werden bidirektionale Streams zum Austausch von Datenpaketen verwendet, die als Input-/Output-Stream gleichermaßen durch Einbeziehung der ausgehandelten Mechanismen geschützt werden.

Für den Verbindungsaufbau wurden Leistungsdaten ermittelt. Folgende Rechnerarchitektur kam zum Einsatz:

1. Rechner: AMD K6 233MHz, 64MByte RAM, Windows NT, Java-VM
2. Rechner: Pentium Pro 200MHz, 64Mbyte RAM,Windows 95, Java-VM
3. Netz: 10 Mbit/s Ethernet, Paketantwortzeiten (Ping) innerhalb Subnetz bei 4096 Bit von 10 ms; über mit LAN-Technologie verbundene Subnetze bei 4096 Bit von 20 ms.

Die Sicherheitsarchitektur wurde für Client und Server der Teleshopping-Applikation konfiguriert. Dabei wurden verschiedene Szenarien kombiniert getestet:

1. Für Aushandlung der Schutzziele: Konfiguration der Schutzziele (Anwendungskonfiguration) so, daß keine Mechanismen verwendet werden müssen, d. h. alle Schutzziele never (Fall 1a) bzw. Konfiguration der Schutzziele so, daß zu jedem Schutzziel ein Mechanismus verwendet werden muß, d. h. alle Schutzziele always (alle Schutzziele verlangen Mechanismus - Fall 1b).

2. Für Aushandlung der Mechanismen: Konfiguration der Mechanismen (Grundkonfiguration) so, daß beide Partner den gleichen und den schnellsten aller Mechanismen am höchsten priorisiert haben (Partner voll übereinstimmend - Fall 2a) bzw. nur der Mechanismus mit der jeweils letzten Priorität übereinstimmend ist und überdies der langsamste Kryptomechanismus ist (Partner schwach übereinstimmend - Fall 2b).

3. Für gesicherte Kommunikation: verschiedene Nachrichtenlängen (1Byte, 1kByte).

Die Tabelle 5 zeigt Meßergebnisse für die Dauer des Aufbaus und der Durchführung einer durch Kryptomechanismeneinsatz gesicherten Kommunikation (Aktion der Applikation). Die Werte stehen in folgendem Zusammenhang untereinander (am Beispiel der vierten Meßwertspalte): 10,21s + (30-1)*0,37s + 6,72s = 27,66s.

Sobald in der Anwendungskonfiguration Schutzziele für ihre Umsetzung Mechanismen verlangen, steigen sowohl die Dauer der Aushandlung als auch die benötigte Zeit für eine Nachricht stark an. Dies ist darauf zurückzuführen, daß sämtliche benötigten Mechanismenklassen instantiiert und Nachrichten mit dem je Schutzziel geforderten Mechanismus bearbeitet werden müssen. Bei den Versuchen einigten sich die Partner im Fall schwach übereinstimmender Grundkonfiguration z.B. beim Schutzziel Vertraulichkeit auf den Mechanismus RSA mit 4096 Bit Schlüssellänge. Dieser wurde von beiden gering priorisiert, war aber der einzige übereinstimmende.

1) Anwendungskonfiguration	a) kein Schutz-ziel verlangt Mechanismus		b) alle Schutzziele verlangen Mechanismus			
2) Grundkonfiguration	-		a) Partner voll überein-stimmend		b) Partner schwach übereinstimmend	
3) Nachrichtengröße	1 Byte	1 kByte	1 Byte	1 kByte	1 Byte	1 kByte
Anzahl der gesendeten Nachrichten	3000	3000	30	30	30	30
Gesamtdauer mit Aushandlung [s]	5,33	5,48	24,98	27,66	58,23	85,51
Aushandlung [s]	0,76		6,72		27,65	
Erste Nachricht [s]	0,17	0,20	14,20	10,21	9,12	10,30
Mittelwert ab 2.Nachricht [s]	<0,01	<0,01	0,14	0,37	0,74	1,64
Maximum ab 2.Nachricht [s]	0,02	0,04	0,16	0,50	0,99	1,69
Minimum ab 2.Nachricht [s]	<0,01	<0,01	0,13	0,31	0,72	1,59

Tabelle 5: Dauer der Aushandlung und der gesicherten Aktion (Mittelwerte über drei Versuche, Kommunikation über mehrere Subnetze hinweg)

Es fällt eine deutlich größere Verweilzeit der jeweils ersten Nachricht gegenüber folgenden auf. Hier werden zunächst Klassen instantiiert. Es zeigt sich, daß für die lokal auszuführenden Aktivitäten (Laden der Java-Klassen, Interpretieren) relativ gesehen wesentlich mehr Zeit benötigt wird als für die eigentliche Übertragung und auch mehr Zeit als für die Sicherung mit Hilfe der Kryptomechanismen bei relativ kurzen Nachrichten. Hier besteht Optimierungsbedarf.

4 Zusammenfassung und Ausblick

Der vorliegende Beitrag gab einen Überblick über die SSONET-Architektur zur Realisierung flexibler mehrseitiger Sicherheit. Als wesentliche Eigenschaften wurden die Mechanismen zur Aushandlung von Sicherheitsanforderungen sowie zur Konfigurierung entsprechender Sicherheitsmechanismen vorgestellt. Mit dem zusätzlichen Konzept der Sicherheitsgateways lassen sich auch bei grundsätzlich inkompatiblen Sicherheitsanforderungen Kommunikationsvorgänge unter Wahrung von Sicherheitseigenschaften realisieren.

Die vorgestellten Ansätze wurden vollständig realisiert und am Beispiel einer Electronic-Commerce-Anwendung validiert. Im Gegensatz zu existierenden Ansätzen wie CORBA Security, Secure Socket Layer oder verschiedenen Sicherheitsplattformen zeichnet sich SSONET gerade durch die verstärkte Flexibilisierung bei der Nutzung von Sicherheitsmechanismen aus.

Im Rahmen weiterführender Arbeiten ist geplant, die SSONET-Konzepte direkt mit konkreten Systemlösungen und Produkten im Umfeld von Internet-Anwendungen zu integrieren. Aus konzeptioneller Sicht sollen die Ansätze zu Sicherheitsgateways weiter verfeinert und formalisiert werden. Auch verschiedene Weiterentwicklungen

der Aushandlungsprotokolle sind vorgesehen. Durch die Einführung von Aktionsklassen für bestimmte Anwendungsabläufe mit einheitlichen Sicherheitseigenschaften soll ferner ein noch höheres Abstraktionsniveau bei der Spezifikation von Sicherheitsanforderungen angeboten werden.

Literatur

BKMS_96 I. Biehl, H. Kenn, B. Meyer, J. Schwarz, C. Thiel: LISA - Library for Secure Applications. Universität des Saarlandes, 1996

FePf_97 H. Federrath, A. Pfitzmann: Bausteine zur Realisierung mehrseitiger Sicherheit. in: G. Müller, A. Pfitzmann (Hrsg.): Mehrseitige Sicherheit in der Kommunikationstechnik, Addison-Wesley-Longman 1997, 83-104

GaPS_98 G. Gattung, U. Pordesch, M.J. Schneider: Der mobile persönliche Sicherheitsmanager. GMD Report 24, GMD-Forschungszentrum Informationstechnik GmbH, 1998

GeKo_95 M. Gehrke, E. Koch: A Security Platform for Future Telecommunication Applications and Services. In Proc. of the 6th Joint European Networking Conference (JENC6), Israel, 1995, http://www.igd.fhg.de/www/igd-a8/pub/pub.html

Kann_94 R. Kanne: Eine objektorientierte Klassenbibliothek für kryptographische Systeme. Diplomarbeit, Universität Hildesheim, 1994

Kran_96 A. Krannig: PLASMA - Platform for Secure Multimedia Applications. In Proceedings Communications and Multimedia Security II, Essen, 1996

OMG_97 Object Management Group: CORBA Services; Updated Revised Edition; OMG Document 97-12-02, Framingham, USA, 1997

PSWW_98 A. Pfitzmann, A. Schill, A. Westfeld, G. Wicke, G. Wolf, J. Zöllner: A Java-Based Distributed Platform for Multilateral Security. In: W. Lamersdorf, M. Merz (Hrsg.): Trends in Distributed Systems for Electronic Commerce, TREC'98, Lecture Notes of Computer Science (LNCS) 1402, Springer, Berlin 1998, 52-64

SECU_98 SECUDE Overview.http://saturn.darmstadt.gmd.de/TKT/security/secude/

SSL_97 A. O. Freier, P. Karlton, P. C. Kocher: The SSL Protocol Version 3.0, INTERNET-DRAFT. http://www.consensus.com/ietf-tls/tls-ssl-version3-00.txt Tansport Layer Security Working Group

TLS_97 T. Dierks, C. Allen: The TLS Protocol Version 1.0, INTERNET-DRAFT. http://www.consensus.com/ietf-tls/tls-protocol-03.txt. Transport Layer Security Working Group