Vertrauen als integraler Bestandteil kryptografischer Spezifikationen

Matthias Schunter
Uni Hildesheim, Institut für Informatik
Marienburger Platz 22, 31141 Hildesheim
schunter@informatik.uni-hildesheim.de

Zusammenfassung

Jedes kryptografische System basiert auf einem Vertrauensmodell (Struktur und vertraute Teile). Diese Arbeit skizziert Mittel zur *expliziten* Angabe von Vertrauen für kryptografische Systeme.

1 Einleitung

Jedes kryptografische Verfahren basiert auf einem Vertrauensmodell. Meist wird es jedoch nicht oder nur implizit angegeben, obwohl es bei Spezifikation und Evaluierung von mehrseitig sicheren kryptografischen Systemen benötigt wird.

Ziel dieser Arbeit ist, die Notwendigkeit expliziter Vertrauensmodelle für mehrseitig sichere kryptografische Systemen zu verdeutlichen und geeignete Formalismen zur kryptografischen Spezifikation von Integritäts- und Verfügbarkeitseigenschaften zu skizzieren.

Prinzipiell verstehen wir unter einem *Vertrauensmodell* die Angabe einer *Struktur* (Komponenten und Verbindungen) und der vertrauten Komponenten. Dies kann sowohl auf Benutzer, als auch auf Systemkomponenten angewendet werden. Unser Ansatz basiert auf einer stufenweisen Verfeinerung der Spezifikation: Zuerst wird der zu erbringende Dienst als Liste von *Dienstanforderungen*[1] spezifiziert. Dann wird das Vertrauensmodell für die Benutzer und anschließend das Vertrauensmodell für die internen Systemkomponenten angegeben. Schließlich wird ein gegebenes Verhalten der internen Systemkomponenten evaluiert.

Ein einfaches Beispiel ist das Vertrauensmodell eines Systems zur Datenübertragung. Es gibt zwei Benutzer „Sender" und „Empfänger". Die Dienstanforderungen sind Integrität (eine übertragene Nachricht wird nicht verändert) und Verfügbarkeit (eine vom Sender eingegebene Nachricht wird übertragen). Eine mögliche Systemspezifikation ist je ein vertrauter Prozeß zur Ver- und Entschlüsselung und ein Netzwerk. Das Vertrauensmodell zu dieser Dienst- und Systemspezifikation ist: Beide Benutzer und Benutzerprozesse sind vertraut. Dem Netzwerk wird nur bezüglich Verfügbarkeit, nicht jedoch bezüglich Integrität vertraut. An diesem einfachen Beispiel wird bereits deutlich, daß das Vertrauen anforderungsbezogen ist und daß das Vertrauensmodell bei der Evaluierung eine

[1] Als Worte einer Spezifikationssprache.

wesentliche Rolle spielt: Würde dem Netzwerk bezüglich Integrität vertraut, so müßten keine integritätssichernden Maßnahmen getroffen werden.

Es existieren bisher keine einheitlichen formalen Definitionen oder Spezifikationsmechanismen für reaktive kryptografische Systeme mit Berücksichtigung des Vertrauensmodells. Allgemeine Definitionsansätze wurden halbformal für Signatursysteme in [12] und allgemeiner in [13] vorgeschlagen.

Bisherige Ansätze für Vertrauensmodellierung beschreiben die Ableitung von Komponentenvertrauen aus einem Intialvertrauen [3, 1] oder beschreiben Anforderungen an einzelne Systeme, wobei das verwendete Vertrauensmodell nur implizit angegeben wird [7, 6, 8]. Wir nehmen jedoch das Vertrauen in die einzelnen Komponenten als gegeben an. Die o.g. Ansätze können jedoch bei der Herleitung dieses Vertrauens hilfreich sein.

Wir werden nun zuerst den herkömmlichen Spezifikationsmechanismus kurz skizzieren und dann die kryptografische Spezifikation mit Vertrauensmodell. Anschließend wird beschrieben, wie die kryptografische Spezifikation und das Vertrauensmodell in eine herkömmliche Spezifikation umgesetzt wird.

2 Spezifikationssprache

Für die Spezifikation kryptografischer Protokolle wird eine Spezifikationssprache mit probabilistischer Semantik benötigt. Da die Ideen der Spezifikation weitgehend unabhängig von der verwendeten Sprache sind, können sowohl synchrone probabilistische [14] als auch asynchrone probabilistische Sprachen, wie probabilistic LOTOS [10], timed probabilistic LOTOS [2] oder probabilistisches CSP [11] verwendet werden. Für nichtprobabilistische Aussagen ist auch temporale Logik ausreichend.

Im folgenden wird eine einfache Sprache mit probabilistischer synchroner Semantik verwendet [14, 9]. Die zeitdiskrete synchrone Sprache beschreibt die Struktur einer Komponente Σ_i als eine Menge von Ein- und Ausgabekanälen In und Out, Operationen Γ und eine Menge von Zeitpunkten $\{1, 2, \ldots\}$. Für Komposition steht ein Kompositionsoperator $\bowtie_+$ und ein Restriktionsoperator $\lceil$ zur Verfügung.

Die Semantik definiert das Ausgabeverhalten $\mathcal{O}$ durch eine Folge von Verteilungsfunktionen $\mathcal{O}_t$, die bei gegebenen Ein- und Ausgaben bis zum Zeitpunkt t die Wahrscheinlichkeit aller Ausgaben zum Zeitpunkt $t+1$ angibt. Die Komposition leitet Ausgaben eines Ausgabekanals $c \in Out$ beim nächsten Schritt zu gleichnamigen Eingabekanälen weiter. Der Sichtoperator reduziert die sichtbare Kanalmenge. Analog könnte die Semantik der Sprache auch mit synchronen probabilistischen endlichen Automaten definiert werden.

3 Dienstklassen

Wir beschreiben nun die Spezifikation der erwünschten Dienste eines Systems. Die kryptografische Dienstspezifikation besteht aus Dienstanforderungen an der Benutzerschnittstelle und einem Vertrauensmodell für die Benutzer.

Die Trennung zwischen externem Dienst an der Benutzerschnittstelle, interner Struktur und internem Verhalten dient der Vereinheitlichung der Spezifikation: Der externe Dienst kann für eine Klasse von Systemen einheitlich sein, obwohl sich der interne Aufbau stark unterscheidet. Um dies zu verdeutlichen, werden die Systeme mit äquivalenter Dienstspezifikation als *Dienstklasse* bezeichnet. Die Spezifikation von Dienstklassen legt weder interne Struktur noch internes Verhalten, sondern nur das Gesamtverhalten nach außen fest. Beispiele für Dienstklassen sind Zahlungssysteme oder Datenübertragungssysteme.

3.1 Benutzerschnittstelle und Dienstanforderungen

Sei $C := In \cup Out$ die Kanalmenge der Benutzerschnittstelle des Systems. Einzelne Dienstanforderungen G_i an das Verhalten an dieser Schnittstelle werden als Worte der Spezifikationssprache ausgedrückt. Beispiele sind Übertragung einer Nachricht oder erfolgreiche Signatur (ausführlicher in [12]).

3.2 Vertrauensmodell für die Benutzer

Jeder Benutzer kontrolliert eine Teilmenge U_i der Kanäle C der Benutzerschnittstelle I, wobei Ausgabekanäle mehrfach vorkommen dürfen.

Nach Festlegung der Dienste und der Benutzer werden nun zu jeder Dienstanforderung G_i *Interessensgruppen* $V_i \subseteq U$ angegeben. Die Benutzer dieser Interessengruppe haben G_i als gemeinsame Dienstanforderung und vertrauen sich für diese Anforderung. Wir modellieren das Vertrauen für jedes Teilziel zunächst binär, d.h. für jedes einzelne Teilziel wird angegeben, ob einem Benutzer oder einer Komponente vertraut wird oder nicht.

Die kryptografische Spezifikation des externen Verhaltens ist also

$$V := \langle I, U, ((G_1, V_1), \ldots, (G_m, V_m)) \rangle \, .$$

Um die kryptografische Spezifikation externen Verhaltens zu verdeutlichen, wird nun eine Anforderung des Datenübertragungssystems formaler beschrieben. Die Benutzerschnittstelle ist $\langle \{sender\}, \{receiver\}, \{\texttt{send, receive}\} \times \Gamma^*, \{1, 2, \ldots\}, \rangle$ für ein Nachrichtenalphabet Γ. Eine Formalisierung der Integritätsanforderung in temporaler Logik ist

$$\underset{m}{\forall} \texttt{receive } m \Rightarrow \diamondminus \texttt{send } m,$$

wobei $\diamondminus P$ bedeutet, daß das Prädikat P früher einmal wahr war. Die Interessengruppe zu dieser Dienstanforderung ist $\{\{sender\}, \{receiver\}\}$.

4 Kryptografische Systemklassen

Bisher wurde durch eine Dienstspezifikation der gewünschte externe Dienst spezifiziert. Für kryptografische Systeme muß dies jedoch verfeinert werden, da sich verschiedene kryptografische Systeme bei gleichem Dienst teilweise wesentlich im zugrundeliegenden Vertrauen unterscheiden. Hierzu wird die Dienstspezifikation durch ein Vertrauensmodell für die internen Komponenten des Systems ergänzt. Es wird also eine interne Struktur und für jede Dienstanforderung das Vertrauen in die internen Komponenten angegeben. Die Spezifikation einer kryptografischen Systemklasse bestehend aus Dienstspezifikation und Vertrauensmodell für die internen Komponenten wird als *Systemklassenspezifikation* bezeichnet. Systeme mit einheitlicher Systemklassenspezifikation werden in *Systemklassen* zusammengefasst. Beispiele kryptografischer Systemklassen sind public-key Verschlüsselungssysteme oder Signatursysteme.

4.1 Interne Systemstruktur

Die interne Struktur des Systems wird durch Systemkomponenten und deren Verbindungen vorgegeben.

Das Datenübertragungssystem besitzt die internen Komponenten Sendergerät `S`, Empfängergerät `E` und Netzwerk `N`. Abb. 1 zeigt Benutzerschnittstelle und Komponenten des Systems.

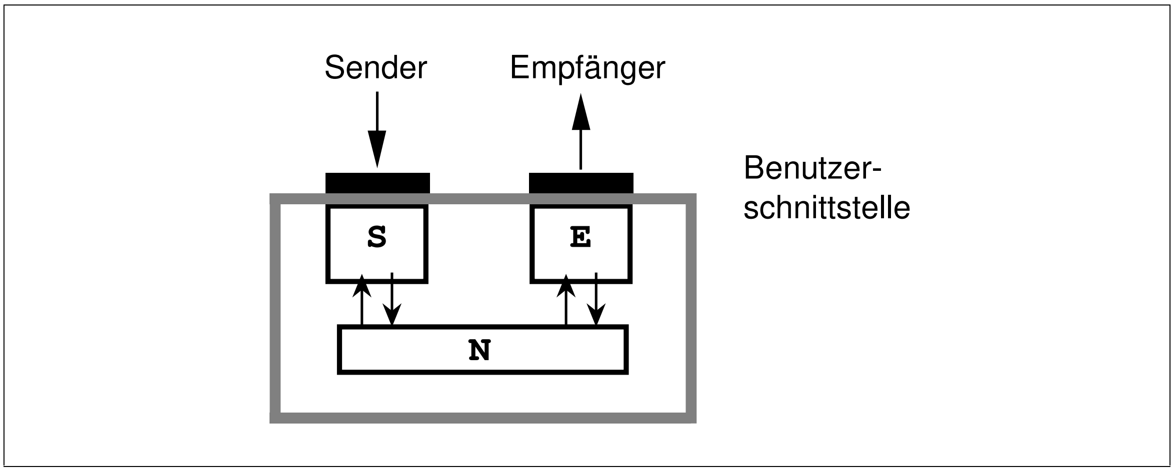

Abbildung 1: Benutzerschnittstelle und Komponenten eines Datenübertragungssystems

4.2 Vertrauen in Systemkomponenten

Analog zum Vertrauen in die Benutzer wird nun für jede Anforderung G_i die Teilmenge der Systemkomponenten V_i' angegeben, denen bezüglich dieser Anforderung vertraut wird. In vielen Fällen kann man das Vertrauen in die Systemkomponenten vom Vertrauen in die Benutzer ableiten (vgl. Abschnitt 7). Die Angabe der vertrauten Komponenten für jede einzelne Dienstanforderung wird als *Systemvertrauen* bezeichnet.

Bei dem Datenübertragungssystem wird Sender- und Empfängergeräten bezüglich Integrität und zusätzlich dem Netzwerk bezüglich Verfügbarkeit vertraut.

Das Vertrauen in die Systemkomponenten spiegelt das Vertrauen in eine korrekte Implementierung wieder: Es wird angenommen, daß vertraute Komponenten gemäß ihrer Spezifikation arbeiten.

Wird einer Komponente also für keine der Dienstanforderungen vertraut, so muß ihr Verhalten auch nicht spezifiziert werden.

5 System

Um ein einzelnes *System* zu spezifizieren, wird die Spezifikation der Systemklasse um eine Spezifikation des internen Verhaltens der Komponenten ergänzt. Das Verhalten der einzelnen Komponenten wird entweder durch eine herkömmliche Spezifikation oder durch Angabe einer Implementierung festgelegt. Wird von einigen Komponenten mehrseitige

Sicherheit gefordert, so kann der gesamte Prozeß für diese Komponenten wiederholt werden. Die Evaluierung, ob die Implementierung einer Komponente der Spezifikation der Komponente entspricht, wird hier nicht behandelt.

6 Kryptografische Semantik

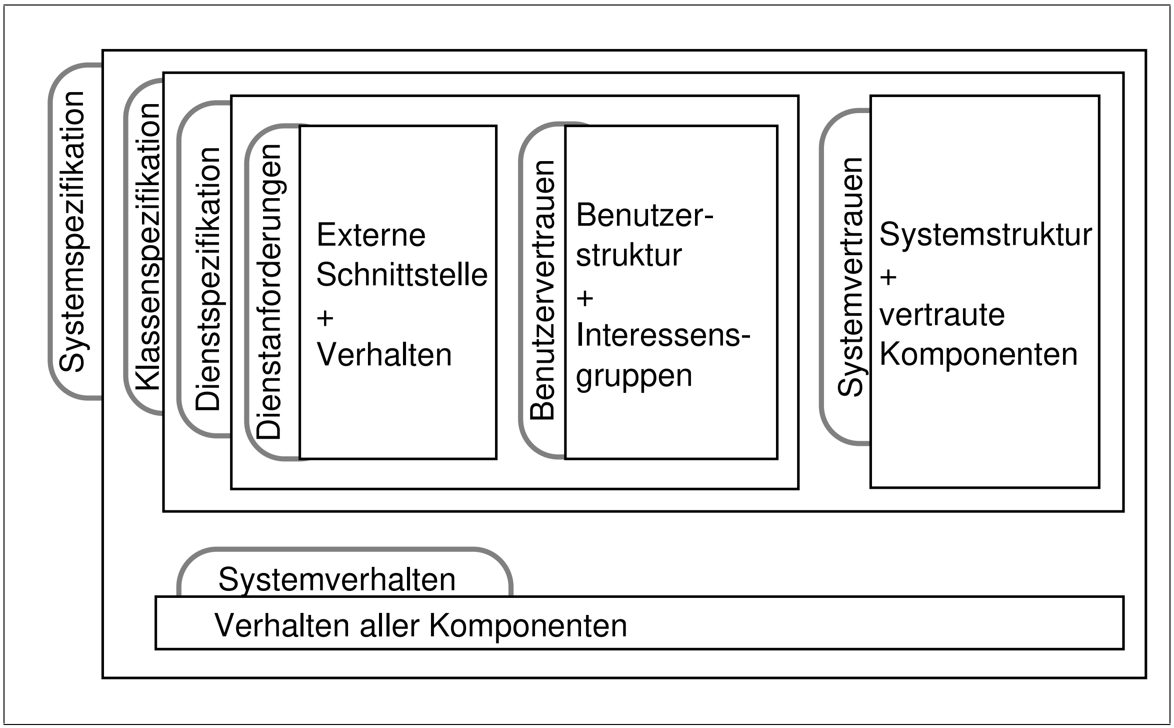

Abbildung 2: Begriffsstruktur im Überblick

Zusammenfassend zeigt Abbildung 2 die Struktur des errichteten Begriffsgebäudes.
Bei der Evaluierung sind zwei Fragestellungen zu unterscheiden:

1. Erfüllt ein System gemäß Systemspezifikation die Dienstspezifikation?
2. Erfüllt eine Implementierung die Systemspezifikation?

Frage 2 wird durch die formale Semantik der Spezifikationssprache beantwortet.
Um zu prüfen, ob ein System gemäß Systemspezifikation den gewünschten Dienst erbringt, wird die Systemspezifikation bestehend aus Dienst-, Systemklassen- und Verhaltensspezifikation in eine herkömmliche Spezifikation umgesetzt. Bei der Evaluierung ist zu beachten, daß für jede Dienstanforderung nur Struktur und Verhalten von vertrauten Systemkomponenten als festgelegt angenommen werden können. Bei nichtvertrauten Systemteilen wird nicht einmal die Struktur beibehalten. Die Systemspezifikation wird für jede Dienstanforderung G_i wie folgt umgesetzt:

1. Die Verhaltens- und Strukturspezifikation der vertrauten Systemkomponenten wird übernommen.

2. Die Strukturspezifikation der Interessensgruppe wird übernommen. (Falls das System nicht robust gegen Fehlbedienung sein muß, kann bestimmtes Benutzerverhalten als Voraussetzung spezifiziert werden.)
3. Die nichtvertrauten Teile (Systemteile und Benutzer) werden als „Angreifer“ zusammengefasst. Weder Verhalten noch Struktur können hier festgelegt werden.

Anschließend wird geprüft, ob diese Spezifikation die Dienstanforderung G_i impliziert.

7 Vertrauensmodelle

In diesem Abschnitt beschreiben wir einige ausgewählte Standardvertrauensmodelle, die die Ableitung des Vertrauens in Systemkomponenten aus gegebenem Vertrauen in die Benutzer vereinfachen.

7.1 Minimales Vertrauen

Bei diesem Vertrauensmodell vertrauen die Benutzer der Interessengruppe nur sich selbst und einer direkt angeschlossenen Systemkomponente (Modell des eigenen Rechners oder eines portablen Benutzerendgeräts). In der Regel sind diese Systemkomponenten durch ein nichtvertrautes Netzwerk verbunden.

Beispiele sind vertrauliche Speicherung von Daten (ein Benutzer) oder vertrauliche Kommunikation (zwei Benutzer).

7.2 Trust Center

Zusätzlich zum Minimal Trust vertrauen hier alle Benutzer für einzelne Teilziele einem Trust Center. Ein Beispiel sind Schlüsselzertifikationsinstanzen bei Signaturverfahren, denen bzgl. Authentizität der Schlüssel, nicht aber bzgl. Authentizität der Nachrichten oder Vertraulichkeit der Schlüssel vertraut werden sollte. Es soll also für das Trust Center nicht möglich sein, Signaturen von einzelnen Benutzern zu fälschen.

7.3 Wallet-Observer Trust

Bisher wurde die Struktur der nichtvertrauten Teile nicht eingeschränkt. Das Wallet-Observer Vertrauensmodell [5, 4] ist ein Beispiel für ein Vertrauensmodell, welches die Struktur des Angreifers einschränkt: Die Anforderung „Unverkettbarkeit der Zahlungen“ für die Benutzer muß nur erfüllt werden, solange der Observer strukturell vom Restangreifer getrennt ist.

Danksagung

Ich danke Birgit Pfitzmann, Gerrit Bleumer und Michael Waidner für zahlreiche Diskussionen und Anregungen zu Vertrauen in kryptografischen Systemen.

Literatur

[1] Thomas Beth, Malte Borcherding, Birgit Klein. Valuation of Trust in Open Networks. ESORICS 94 (Third European Symposium on Research in Computer Security), Brighton, LNCS 875, Springer-Verlag, Berlin 1994, 3-18.

[2] Ed Brinksma, Joost-Pieter Katoen, Rom Langerak, Diego Latella. Performance Analysis and True Concurrency Semantics. University of Twente, university of technical and social sciences, Tele-Informatics and Open Systems Group, Department of Computer Science and Department of Electrical Engineering, June 27th, 1994,1-28.

[3] Michael Burrows, Martin Abadi, Roger Needham. A Logic of Authentication. Operating Systems Review 23/5 (1989) 1-13.

[4] David Chaum. Achieving Electronic Privacy. Scientific American (August 1992) 96-101.

[5] David Chaum, Torben Pryds Pedersen. Wallet Databases with Observers. Crypto '92, LNCS 740, Springer Verlag, Berlin 1993, 89-105.

[6] Zvi Galil, Stuart Haber, Moti Yung. Security against Replay Chosen-Chipertext-Attack. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 2:183–189, 1991.

[7] Oded Goldreich. A Uniform-Complexity Treatment of Encryption and Zero-Knowledge. *Journal of Cryptology*, 6:21–53, 1993.

[8] Shafi Goldwasser, Silvio Micali, Ronald L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput. 17/2 (1988) 281-308.

[9] James W. Gray III. Toward a Mathematical Foundation for Information Flow Security. Journal of Computer Security 1/3,4 (1992) 255-294.

[10] Joost-Pieter Katoen, Rom Langerak, Diego Latella. Modeling Systems by Probabilistic Process Algebra: An Event Structures Approach. Participants' proceedings FORTE'93, Sixth international conference on Formal Description Techniques, Boston, Massachusetts, USA, October 26-29, 1993.

[11] Kim G. Larsen, Arne Skou. Compositional Verification of Probabilistic Processes. Concur '92, LNCS 630, Springer-Verlag, Berlin 1992, 456-471.

[12] Birgit Pfitzmann. Sorting Out Signature Schemes. 1st ACM Conference on Computer and Communications Security, 3.-5.11.1993, Fairfax, acm press 1993, 74-85.

[13] Birgit Pfitzmann, Michael Waidner. A General Framework for Formal Notions of "Secure"System. Hildesheimer Informatik-Berichte 11/94, ISSN 0941-3014, Institut für Informatik, Universität Hildesheim, April 1994.

[14] Matthias Schunter. Spezifikation vom Geheimhaltungseigenschaften für reaktive kryptologische Systeme. Diplomarbeit, Universität Hildesheim, Institut für Informatik, Hildesheim, Jan 1994.