Vertrauenswürdiges Booten als Grundlage authentischer Basissysteme

Michael Groß
Gesellschaft für Mathematik und Datenverarbeitung (GMD)
Darmstadt

Zusammenfassung

Für viele Anwendungen von Computersystemen benötigt man ein vertrauenswürdiges Basissystem als Grundlage für die Garantie bestimmter Funktionalitäten des Gesamtsystems. Vertrauenswürdiges Booten ist ein Verfahren, ein bestimmtes Betriebssystem auf einer gegebenen Hardware so zu starten, daß die Authentizität beider Komponenten überprüft werden kann. Wenn wir eine vertrauenswürdige Zentraleinheit und ein vertrauenswürdiges Betriebssystem voraussetzen, dann kann man beide so zusammenfügen, daß ein vertrauenswürdiges Basissystem entsteht, welches in der Lage ist, sich anderen Instanzen gegenüber zu authentifizieren.

1 Einleitung

Das Vertrauen eines Anwenders in ein System beruht auf seinem Vertrauen in die einzelnen Systemkomponenten. Da die Sicherheit eines Betriebssystems oder der Zentraleinheit im allgemeinen nicht überprüfbar ist, ist der Anwender darauf angewiesen, sich auf die Zuverlässigkeit der Hersteller zu verlassen. Dies ist natürlich nur möglich, wenn diese Systemkomponenten authentisch sind, sie nicht manipuliert wurden und ihre Hersteller bekannt sind.

Eine Workstation eines vertrauenswürdigen Herstellers ist nur dann vertrauenswürdig, wenn sie nachweislich nicht manipuliert wurde und von einem sicheren Betriebssystem kontrolliert wird. Um ein sicheres verteiltes System zu erhalten, ist es deshalb notwendig, die beteiligten Knoten und ihre Betriebssysteme zu authentifizieren. Nur unter dieser Voraussetzung kann das Vertrauen eines Anwenders in die Sicherheit des Gesamtsystems auf die Zuverlässigkeit der Hersteller zurückgeführt werden.

Da verteilte Systeme aus einem Netzwerk mit vielen Knoten bestehen, ergeben sich zur Durchsetzung der Sicherheitsinteressen eines Anwenders spezielle Probleme. Insbesondere bei der Kommunikation von Knoten untereinander, jedoch auch bei der Kommunikation

des Anwenders mit dem System, ist die Authentizität der beteiligten Rechnerknoten eine Voraussetzung für die Sicherheit des Gesamtsystems. Im vorliegenden Papier wird ein Verfahren beschrieben, das es erlaubt, beim Bootvorgang die Voraussetzungen zur Authentifizierung von Knoten und Software und zum Aufbau von sicheren Kommunikationskanälen zu schaffen.

In vielen verteilten Systemen werden als Voraussetzung für die Systemsicherheit unmanipulierbare Knoten und ein authentischer Betriebssystemkern postuliert. In der Praxis sind die geforderten Voraussetzungen jedoch schwer durchzusetzen, da eine Manipulation an der Hardware normalerweise nur mit hohem Aufwand (z.B. spezielle, gesicherte Maschinenräume) auszuschließen ist. Auch der Betriebssytemcode darf nicht ausgetauscht werden können, was in der Praxis wohl zumindest eine für Angreifer unzugängliche Aufbewahrung aller Betriebssytemdatenträger erforderlich macht.

Da Manipulationen jedoch niemals vollständig ausgeschlossen werden können, ist es wünschenswert, Veränderungen an den Knoten eines verteilten Systems erkennbar zu machen. Sowohl der Benutzer als auch andere Knoten im Netz müssen eine Möglichkeit haben, die Authentizität und Integrität jedes beteiligten Knotens jederzeit zu überprüfen. Weiterhin soll in einem Netz mit privaten Knoten jedes beliebiege Betriebssystem gebootet werden dürfen, ohne die Sicherheit der anderen Knoten zu beeinträchtigen. Dies erfordert, daß die Authentizität eines Betriebssystems sowohl vom Benutzer als auch von anderen Knoten während des Betriebs überprüft werden kann.

1.1 Authentische Systemkomponenten

Zu den Aufgaben eines sicheren Betriebssystems gehört die Prüfung der Authentizität von Anwendungssoftware. Auf eine solche Prüfung kann sich der Anwender jedoch nur verlassen, wenn das Betriebssystem selbst authentisch ist und die Annahmen des Betriebssystemherstellers über die Funktionalität der Hardware stimmen. Heute übliche Computersysteme bieten jedoch keine ausreichenden Möglichkeiten zur Überprüfung der Hardware und des Betriebssystems.

Die Prüfung der Authentizität einzelner Komponenten eines Systems kann durch andere Systemkomponenten erfolgen, die selbst auch wieder von Komponenten des Systems überprüft werden. Auf diese Weise entsteht eine Kette von Authentizitätsprüfungen, wobei am Anfang einer solchen Prüfungskette eine Komponente steht, die nicht innerhalb des Systems selbst überprüft werden kann. Die Authentizität dieser Komponente muß deshalb von einer externen Instanz (z.B. dem Anwender) geprüft werden können.

Die Systemkomponente am Anfang der Prüfungskette sollte ein Teil der Hardware sein, dessen Integrität mit herkömmlichen Methoden so gesichert wird, das Integritätsverletzungen erkennbar werden (z.B. eingießen in Kunststoff). Bei unverletzter Integrität müßte diese Komponente in der Lage sein, sich anderen Instanzen gegenüber zu authentifizieren und einen Betriebssystemkern so zu laden, daß auch dieser sich gegenüber anderen Instanzen authentifizieren kann.

Die Zentraleinheit eines Computersystems, bestehend aus Prozessor, Bus, Hauptspeicher und E/A-Kanälen, ist eine für diese Anforderungen geeignete Systemkomponente, da ihr

Schutz vor externen Bedrohungen (z.B. Abhören des Busses muß unmöglich sein) von vielen Sicherheitsmechanismen der Software sowieso vorausgesetzt wird und bis auf einige kleinere Erweiterungen der Zentraleinheit für unsere Zwecke keine weiteren Änderungen an der Hardware nötig werden.

Ausgehend von einem unmanipulierten und authentifizierbaren Betriebssystemkern auf einer unmanipulierten und authentifizierbaren Zentraleinheit kann das Betriebssystem dann seine weiteren Bestandteile selbst überprüfen. Eine wichtige Voraussetzung für das Vertrauen des Anwenders und der Hersteller von Anwendungssoftware in die Sicherheit eines Betriebssystems wäre damit erfüllt.

In Verbindung mit dem Betriebssystem bildet die Zentraleinheit ein Basissystem für die oben beschriebene Prüfungskette, welches in der Lage ist, sich gegenüber Dritten zu authentifizieren, und dessen Vertrauenswürdigkeit unter der Voraussetzung zuverlässiger Hersteller gewährleistet ist kann. Dieses Basissystem kann nicht nur als Fundament für alle weiteren Prüfungen der Prüfungskette dienen, sondern ist auch allgemein als Basis für verschiedene Sicherheitsmechanismen des Betriebssystems und der Anwendungsprogramme geeignet.

Ein Basissystem entsteht beim Bootvorgang durch das Starten eines Betriebssystems auf einer Zentraleinheit. Dabei wird vom Urlader zunächst ein Teil des Betriebssystems in den Hauptspeicher der Zentraleinheit geladen, welcher anschließend die Kontrolle über den weiteren, betriebssystemspezifischen Ablauf des Bootvorgangs übernimmt. Wenn der Urlader vor Übergabe der Kontrolle die Authentizität dieses Betriebssystemteils prüft und die Zentraleinheit physikalisch vor unbemerkbaren Veränderungen geschützt ist, kann man ein vertrauenswürdiges Betriebssystem auf einer vertrauenswürdigen Zentraleinheit so booten, daß ein vertrauenswürdiges Basissystem entsteht.

1.2 Vertrauenswürdige Zentraleinheiten

Die Funktionalität der Zentraleinheit (ZE) kann nur dann gewährleistet werden, wenn die Zentraleinheit authentisch ist und nicht manipuliert wurde. In diesem Falle hängt die Vertrauenswürdigkeit der Zentraleinheit von den Fähigkeiten und der Zuverlässigkeit ihres Herstellers ab. Der Anwender eines Computersystems muß deshalb jederzeit feststellen können, ob die Zentraleinheit manipuliert wurde oder nicht. Dazu gibt es prinzipiell drei Möglichkeiten:

- Er vertraut auf die Prüfungen eines Systemverwalters.
- Er prüft die Unversehrtheit und Authentizität des Gehäuses der Zentraleinheit.
- Er besitzt spezielle Hardware zum Prüfen (z.B. Chipkarte) und die Zentraleinheit kann sich gegenüber dieser Hardware authentifizieren.

Für die Vertrauensürdigkeit einer Zentraleinheit ist ihre Authentizität alleine nicht ausreichend. Sie ist jedoch eine wichtige Voraussetzung. Auf eine Überprüfungsmöglichkeit kann daher keinesfalls verzichtet werden.

1.3 Vertrauenswürdige Betriebssysteme

Die Sicherheit eines Betriebssystems wird vom Betriebssystemhersteller spezifiziert und der Benutzer ist darauf angewiesen, dem Hersteller zu vertrauen, daß das Betriebssystem diese Spezifikation voll erfüllt. Ebenso wie bei der Zentraleinheit muß deshalb auch bei einem vertrauenswürdigen Betriebssystem jederzeit der Hersteller feststellbar sein. Ebenso muß überprüft werden können, ob der Betriebssystemcode nach der Auslieferung manipuliert wurde. Nur so kann das Vertrauen in ein Betriebssystem (BS) auf dem Vertrauen in die Fähigkeiten und die Zuverlässigkeit des Betriebssystemherstellers beruhen.

Eine mögliche Sicherung des Betriebssystemcodes liegt auf der Hand: Damit das Betriebssystem vom Binden beim Hersteller bis zum Booten beim Kunden nicht mehr unbemerkt verändert werden kann, signiert der Hersteller sein Betriebssystem vor der Auslieferung mit Hilfe eines kryptografischen Signierverfahrens. Dies garantiert, daß Manipulationen am Betriebssystemcode erkennbar werden.

1.4 Welches Betriebssystem läuft im Moment in der ZE?

Forderung: Der Anwender eines Systems muß dem System vertrauen können.

Um diese Forderung zu erfüllen gibt es zwei Möglichkeiten:

- Er vertraut darauf, daß es einen zuverlässigen Systembetreuer gibt, der die Vertrauenswürdigkeit des Systems garantieren kann.
- Er kann selbst kontrollieren, mit welchen Betriebssystemen und Anwenderprogrammen er auf welchen Zentraleinheiten arbeitet, und vertraut den Herstellern.

Hier soll die zweite Möglichkeit betrachtet werden, da der Systembetreuer ebenfalls als Anwender betrachtet werden kann, der sich zunächst von der Authentizität des Basissystems überzeugen muß.

Die Authentizität einer Zentraleinheit kann vom Anwender nicht immer direkt geprüft werden, da sich die Zentraleinheit nicht unbedingt an seinem Arbeitsplatz befinden muß. Es sollte also möglich sein, ihre Authentizität mit kryptographischen Protokollen über eine größere Entfernung zu prüfen.

Bei einer Prüfung des Betriebssystems im laufenden Betrieb müßten sowohl der Code als auch die aktuellen Daten auf Konsistenz und Korrektheit überprüft werden, um sicher zu sein, daß das Betriebssystem keine trojanischen Pferde enthält und nicht in unerlaubter Weise manipuliert wurde. Dazu wäre es zumindest nötig, mit Testprogrammen von außerhalb des Betriebssystems auf den Adressraum des Betriebssystems zugreifen zu können. Dies stellt jedoch generell ein erhebliches Sicherheitsrisiko dar.

Aus diesen Gründen sollte die Überprüfung des Betriebssystems beim Bootvorgang durchgeführt werden, d.h. der Anwender oder zumindest der Systembetreuer muß dann nur noch überprüfen können, welches Betriebssystem beim letzten Reset gebootet wurde. Unter der Voraussetzung, daß ein Betriebssystem nicht durch ein anderes ersetzt werden kann und auf einer sicheren Zentraleinheit nicht manipulierbar ist, kann man so jederzeit feststellen, welches Betriebssystem gerade läuft.

2 Kryptographische Methoden

Für die Realisierung des Vertrauenswürdigen Bootens sollen kryptographische Signaturverfahren angewendet werden. Die vorgeschlagene Lösung basiert auf dem Prinzip der digitalen Signatur, das zuerst von Diffie und Hellman vorgestellt wurde [DH76]. Eine Eigenschaft dieser digitalen Signaturen ist ihre Fälschungssicherheit. Dies macht sie auch zu einem geeigneten Instrument für die Authentifizierung von Instanzen.

Die gleichen Verfahren können auch angewendet werden, um die Kommunikation externer Instanzen mit dem von uns betrachteten Basissystem fälschungssicher zu gestalten. Dazu gehen wir davon aus, daß folgende Forderungen zu erfüllen sind:

- Modifikationen an einer Nachricht sind erkennbar.
- Der Absender einer Nachricht kann festgestellt werden.
- Nachrichten werden vor 'message replay' geschützt.

Die konventionellen, klassischen Kryptographieverfahren arbeiten mit einem geheimen Schlüssel, den alle beteiligten Kommunikationspartner kennen müssen. Diese Verfahren nennt man daher symmetrische Verfahren oder Private-Key-Verfahren. Der Schlüssel muß von einem Partner an den anderen oder von einem Dritten an beide übermittelt werden.

Im Gegensatz zu den Private-Key- Verfahren verwenden Public-Key-Verfahren (asymmetrische Verfahren) verschiedene Schlüssel für die Ver- und Entschlüsselung. Der Verschlüsselungsschlüssel (Public-Key, öffentlicher Schlüssel) läßt sich aus dem Entschlüsselungsschlüssel (Secret-Key, geheimer Schlüssel) leicht berechnen und wird veröffentlicht. Die Berechnung des geheimen Schlüssels aus dem öffentlichen Schlüssel ist jedoch unmöglich oder zumindest komplexitätstheoretisch sehr schwer.

Für die Anwendung eines solchen Public-Key-Verfahrens benötigt die Instanz X also das Schlüsselpaar (PK_X, SK_X):

PK = (öffentlicher Schlüssel, Public Key)
SK = (geheimer Schlüssel, Secret Key)

Rivest, Shamir und Adleman haben ein Verfahren vorgestellt [RSA78], das in hohem Maße fälschungssicher zu sein scheint und dem hier verwendeten Signatursystem zugrunde liegen könnte. Die Sicherheit dieses Verfahrens basiert auf dem Problem, eine große natürliche Zahl zu faktorisieren. Die Zerlegung großer Zahlen in ihre Primfaktoren ist aber mit allen zur Zeit bekannten Algorithmen extrem schwierig. Die Ver- und Entschlüsselungszeiten sind trotz softwaremäßiger Optimierung noch recht groß. Für die hier beschriebene Anwendung spielen diese Zeiten jedoch eine untergeordnete Rolle, da nur sehr wenige Ver- und Entschlüsselungen bei jedem Bootvorgang nötig sind.

2.1 Signaturen

Eine digitale Signatur entspricht in ihrer Funktion der Unterschrift einer Person unter einem Dokument. Im Unterschied zu einer herkömmlichen Unterschrift ist eine digitale

Signatur sowohl abhängig von ihrem Erzeuger als auch vom Inhalt des signierten Dokuments. Dies hat zur Folge, daß nach einer Manipulation am signierten Dokument die Signatur nicht mehr zum Dokument paßt und auf diese Weise Manipulationen an digitalen Dokumenten erkennbar werden. Die Instanz, die eine digitale Signatur erzeugen möchte, muß dazu über ein Geheimnis verfügen, welches zur Erzeugung der Signatur benötigt wird. Dies gewährleistet, daß die Signatur ausschließlich von dem Besitzer des Geheimnisses erzeugt werden kann. Eine öffentliche Information erlaubt anderen Instanzen, ohne Kenntnis des Geheimnisses, eine solche Signatur zu verifizieren und so den Inhaber der geheimen Information als Erzeuger der Signatur zu identifizieren. Durch Verifizieren der Signatur wird daher gleichzeitig die signierende Instanz und der Inhalt des Dokuments authentifiziert. Als Signierverfahren könnte hier z.B. der RSA-Algorithmus in Verbindung mit einer Hashfunktion angewendet werden [EHV88] [Zim86].

2.1.1 Signieren einer Nachricht

Mit einer öffentlich bekannten Hashfunktion wird zunächst ein Merkmal fester Länge (Hashwert, MAC, MDC) aus der Nachricht berechnet. Dieser Hashwert wird dann digital unterschrieben – also mit dem eigenen, geheimen Schlüssel SK_X verschlüsselt – und an die Nachricht angehängt.

gegeben:	Nachricht	N
	Secret Key der Instanz X	SK_X
	RSA-Funktion	E
	Hashfunktion	h

1. berechne $h(N)$
2. berechne $E(SK_X, h(N)) = S_{N,X}$
3. konkateniere N und $S_{N,X}$

In folgenden soll die Nachricht stets mit ihrer Signatur gemeinsam betrachtet werden, wobei die Nachricht im Klartext vorliegt und die Signatur angehängt wird.

Notation: $[N]_X$

Signieren einer Nachricht kann also nur die Authentizität dieser Nachricht, nicht jedoch ihre Vertraulichkeit gewährleisten.

2.1.2 Prüfung einer Signatur

Zur Prüfung der Signatur wird nun aus dem Dokument das Merkmal mit Hilfe der Hashfunktion H bestimmt, und dann mit Hilfe des öffentlichen Schlüssels PK die Signatur entschlüsselt. Stimmen Hashwert und entschlüsselte Signatur überein, so stammt die Unterschrift vom Besitzer des geheimen Schlüssels SK_X, und es ist gewährleistet, daß das Dokument nach der Erzeugung der Signatur nicht mehr verändert wurde.

gegeben:	Nachricht mit Signatur	$[N]_X$
	Public Key der Instanz X	PK_X
	RSA-Funktion	D
	Hashfunktion	h

1. berechne $h(N)$
2. berechne $D($ PK_X, Signatur $)$
3. vergleiche $D($ PK_X, Signatur $)$ und $h(N)$

Die Prüfung der Signatur entspricht jedoch nur dann einer Authentizitätsprüfung der Nachricht, wenn der Public Key, der zum Prüfen verwendet wurde, authentisch ist.

2.2 Zertifikate

Eine Instanz, die in der Lage ist, einen geheimen Schlüssel zu speichern und damit digitale Signaturen zu erzeugen, kann sich anderen Instanzen gegenüber dadurch authentifizieren, daß sie eine Nachricht der anderen Instanz unterschreibt und zurückschickt. Voraussetzung für eine solche Authentisierung ist, daß die verifizierende Instanz den öffentlichen Schlüssel der signierenden Instanz kennt, und der geheime Schlüssel dieser Instanz wirklich geheimgehalten wurde. Das Problem der Authentizität des öffentlichen Schlüssels kann man mit Zertifikaten lösen.

Ein Zertifikat ist eine digital signierte Nachricht, mit der eine Zertifizierungsinstanz die Zuordnung von bestimmten Merkmalen zu einer anderen Instanz beglaubigt.

Wird dabei als Merkmal der Besitz eines Schlüsselpaares gewählt, so kann ein Zertifikat die Frage beantworten, zu wem ein öffentlicher Schlüssel gehört, und erlaubt dieser Instanz so, sich anderen Instanzen gegenüber mit Hilfe ihres geheimen Schlüssels selbst zu authentifizieren. Ebenso lassen sich mit Zertifikaten den Inhabern bestimmter Merkmale auch weitere Merkmale zuordnen.

Ein Zertifikat, wie es hier benötigt wird, besteht aus:

a) Merkmale der Instanz
   - Name der Instanz X (Id_X)
   - öffentlicher Schlüssel von X (PK_X)
   - weitere Merkmale von X

b) Signatur der Zertifizierungsinstanz

Notation: $\langle Id_X, PK_X, ... \rangle_{Zertifizierungsinstanz}$

Diese Zertifikate werden von der signierenden Instanz mit der signierten Nachricht mitgeschickt und enthalten ihren Namen und ihren öffentlichen Schlüssel. Für die Authentizitätsprüfung muß die verifizierende Instanz dann nur den öffentlichen Schlüssel der Zertifizierungsinstanz kennen.

3 Sicheres Booten

In konventionellen Systemen wird beim Booten ein Betriebssystem ohne Authentizitätsprüfung von der Zentraleinheit geladen und gestartet. Beim vertrauenswürdigen Booten dagegen soll die Zentraleinheit ein Betriebssystem nicht nur laden und starten, sondern auch vor dem Starten die Authentizität des geladenen Codes prüfen. Weiterhin soll die Zentraleinheit dem Systemverwalter direkt den Namen des geladenen Betriebssystems anzeigen können und dem Betriebssystem ein Schlüsselpaar zur Verfügung stellen, mit dessen Hilfe sich das Betriebssystem gegenüber anderen Systemen authentisieren kann.

Damit der vorgeschlagene Bootmechanismus zu einem vertrauenswürdigen Basissystem führt, müssen sowohl vertrauenswürdige Betriebssysteme als auch vertrauenswürdige Zentraleinheiten zusätzliche Anforderungen erfüllen:

Anforderungen an ein vertrauenswürdiges Betriebssystem:

- Das Betriebssystem ist so konstruiert, daß es nach dem Booten im laufenden Betrieb nicht mehr durch einen anderes Betriebssystem ersetzt werden kann. Es darf keine Möglichkeit geben, den Code oder die Daten zu manipulieren.
- Das Betriebssystem kann einen geheimen Schlüssel sicher aufbewahren und damit digitale Signaturen erzeugen, um sich anderen Systemen gegenüber zu authentifizieren.

Anforderungen an eine vertrauenswürdige Zentraleinheit:

- Die Zentraleinheit kann dem Betriebssystem ein Schlüsselpaar zur Verfügung stellen.
- Die Zentraleinheit kann dem Betriebssystem ein Zertifikat für seinen öffentlichen Schlüssel zur Verfügung stellen, damit sich das Betriebssystem anderen gegenüber mit Hilfe seiner Signatur und des Zertifikats authentifizieren kann.
- Eine Anzeige am Gehäuse der Zentraleinheit zeigt den Namen des geladenen Betriebssystems unfälschbar an und ermöglicht dem Anwender eine direkte optische Kontrolle des Bootvorgangs. Er kann so ohne Zuhilfenahme weiterer (evtl. kompromittierter) Geräte feststellen, welches Betriebssystem die Zentraleinheit kontrolliert.

3.1 Grundprinzipien des vertrauenswürdigen Bootens

Wie kann sich das Betriebssystem authentifizieren?

> Jedes Exemplar eines Betriebssystems besitzt bei der Ausführung ein eigenes Schlüsselpaar. Es kann Authentifizierungsprotokolle durchführen und mit dem geheimen Schlüssel Signaturen erzeugen. Der öffentliche Schlüssel des Betriebssystemexemplars muß dazu veröffentlicht werden.

Wie kann der öffentliche Schlüssel unfälschbar veröffentlicht werden?

> Der öffentliche Schlüssel wird von einer vertrauenswürdigen Instanz zertifiziert, deren öffentlicher Schlüssel authentifizierbar ist.

Woher bekommt das Betriebssystem sein Schlüsselpaar?

> Die Zentraleinheit erzeugt das Schlüsselpaar für das Betriebssystem. Dieses Schlüsselpaar wird dem Betriebssystem beim Booten von der vertrauenswürdigen Zentraleinheit übergeben, nachdem sie sich von der Authentiziät des Betriebssystems überzeugt hat.

Wer zertifiziert den öffentlichen Schlüssel des Betriebssystems?

> Die Zentraleinheit zertifiziert den öffentlichen Schlüssel des Betriebssystems. Dazu besitzt sie ein eigenes Schlüsselpaar und kann damit Signaturen erzeugen.

Woher bekommt die Zentraleinheit ihr Schlüsselpaar?

> Das SchlüSselpaar für die Zentraleinheit wird von ihr selbst erzeugt. Dies geschieht vor der Auslieferung durch den Hersteller, da so der Hersteller den öffentlichen Schlüssel der Zentraleinheit zertifizieren kann.

Für die obigen Antworten gibt es auch Alternativen, die in einem Arbeitspapier der GMD [Gro91] demnächst diskutiert werden sollen. Die prinzipielle Vorgehensweise beim vertrauenswürdigen Bootens bleibt jedoch gleich:

- Jedes Betriebssystemexemplar besitzt ein eigenes Schlüsselpaar.
- Die Zentraleinheit besitzt ebenfalls ein eigenes Schlüsselpaar.
- Die Zentraleinheit ordnet beim Booten dem Betriebssystem sein Schlüsselpaar zu.
- Bei Authentifizierungen des Basissystems werden immer sowohl das Betriebssystem als auch die Zentraleinheit authentifiziert.

3.2 Instanzen

Für das vertrauenswürdige Booten ist es nötig, folgende Instanzen zu unterscheiden, die durch einen weltweit eindeutigen Namen (Id) identifiziert werden können:

- Betriebssystem-Hersteller (BSH)

  Id_{BSH} = (Firmenname, Firmensitz)
  Die Id des Betriebssystemherstellers soll weltweit eindeutig sein. Deshalb sollte nicht nur der Firmenname (national), sondern auch der Firmensitz (international) verwendet werden.

- Betriebssystem (BS)

  Id_{BS} = (BS-Name, Version, Seriennummer, Id_{BSH})
  Jedes auszuliefernde Betriebssystem erhält außer seinem Namen und der Betriebssystemversion eine Seriennummer. Die Id des Herstellers als Bestandteil der Id des Betriebssystems ermöglicht die direkte Identifikation des Herstellers und macht die Id des Betriebssystems weltweit eindeutig.

- Zentraleinheit-Hersteller (ZEH)

  Id_{ZEH} = (Firmenname, Firmensitz)
  Die Id des Herstellers der Zentraleinheit soll weltweit eindeutig sein. Deshalb sollte nicht nur der Firmenname (national), sondern auch der Firmensitz (international) verwendet werden.

- Zentraleinheit (ZE)

  Id_{ZE} = (ZE-Name, Version, Seriennummer, Id_{ZEH})
  Jede produzierte Zentraleinheit erhält außer ihrem Namen und der Versionsnummer eine Seriennummer. Die Id des Herstellers als Bestandteil der Id der Zentraleinheit ermöglicht die direkte Identifikation des Herstellers und macht die Id der Zentraleinheit weltweit eindeutig.

3.3 Bootvorgang

Nach einer Vorbereitungsphase, in der die Zentraleinheit (ZE) und das Betriebssystem (BS) von ihren Herstellern erzeugt worden sind, werden sie beim Bootvorgang zu einem Basissystem kombiniert. Erst in der anschließenden Betriebsphase ist dann die Nutzung des Computersystems durch den Anwender möglich.

Normalerweise wird am Ende des Bootprogramms eine Laderoutine ausgeführt, welche den BS-Lader in den Speicher lädt und startet. Das Problem beim konventionellen Booten besteht darin, das der BS-Lader nicht von der Laderoutine überprüft wird und deshalb unbemerkt modifiziert werden kann. Nach seiner Aktivierung lädt der BS-Lader weitere Teile des BS in den Speicher und initialisiert und startet sie. Dieser Teil des Bootvorgangs ist abhängig vom verwendeten BS.

Der BS-Hersteller kann zwar einen sicheren Mechanismus entwerfen, um das BS gegen unerlaubte Manipulationen im betriebssystemspezifischen Teil des Bootvorgangs zu schützen, aber er ist nicht in der Lage, das Laden des BS-Laders zu kontrollieren. Dieser erste Schritt ist ein Sicherheitsproblem, das nicht vom BS-Hersteller alleine gelöst werden kann und deshalb eine enge Zusammenarbeit mit dem Hersteller der ZE erfordert.

Die vorgeschlagene Methode des vertrauenswürdigen Bootens ermöglicht es der ZE, ein Programm zu laden, seine Authentizität zu prüfen und es zu starten. Ein solches Programm, dessen Code in einem sogenannten Security-Boot-Modul enthalten sein muß, ist der BS-Lader, der dann als Verankerung für die Kette von Sicherheitsmechanismen des BS benutzt werden kann.

Damit die Authentizität des BS-Laders geprüft werden kann, wird der Security-Boot-Modul vom BS-Hersteller unterschrieben. Auf ähnliche Weise müssen sicherlich auch die anderen Teile des BS gesichert werden, was jedoch zu den Sicherheitsmechanismen des BS zählt und deshalb hier nicht betrachtet werden soll.

Security-Boot-Modul	$[Id_{BS}, BS - Lader]_{BSH}$

Während des Bootens bekommt der BS-Lader von der ZE ein Schlüsselpaar und verschiedene Zertifikate, die er an das BS weitergibt. Mit Hilfe des Schlüsselpaares und den Zertifikaten kann sich das BS dann selbsttätig gegenüber anderen Systemen authentifizieren. Durch die Zertifikate wird dabei immer die ZE ebenfalls authentifiziert.

3.4 Die Hersteller

Die Hersteller von Betriebssystemen und Zentraleinheiten besitzen alle eigene Schlüsselpaare. Zur Authentizitätssicherung ihrer öffentlichen Schlüssel gibt es verschiedene Möglichkeiten, die in [Gro91] diskutiert werden sollen. Eine mögliche Lösung ist es, daß ihre öffentlichen Schlüssel von einer internationalen Herstellervereinigung (HV) zertifiziert werden, deren öffentlicher Schlüssel allgemein bekannt ist.

Zertifikat für den Hersteller des BS	$\langle Id_{BSH}, PK_{BSH} \rangle_{HV}$
Zertifikat für den Hersteller der ZE	$\langle Id_{ZEH}, PK_{ZEH} \rangle_{HV}$

3.5 Das Betriebssystem

Jedes aktive Betriebssystemexemplar besitzt ein eigenes Schlüsselpaar, welches von der ZE erzeugt wird. Der geheime Schlüssel (SI(Es) wird beim Booten dem BS von der ZE zur Verfügung gestellt. Das BS ist dann für den weiteren Schutz dieses Schlüssels selbst verantwortlich. Mit dem geheimen Schlüssel kann das BS Signaturen erzeugen. Der zugehörige öffentliche Schlüssel (PK_{BS}) wird von der ZE im Bootzertifikat zertifiziert.

Geheimnis des BS	SK_{BS}
Bootzertifikat	$\langle Id_{BS}, PK_{BS}, PK_{BSH} \rangle_{ZE}$

Da bei jedem Bootvorgang ein neues Schlüsselpaar verwendet wird, ist gewährleistet, daß selbst im Falle der Kompromittierung des geheimen Schlüssels eines Betriebssystemexemplars, nur die Identität dieses einen Exemplars vorgetäuscht werden kann.

3.6 Die Zentraleinheit

Eine vertrauenswürdige ZE muß prinzipiell so geschützt sein, daß niemand unkontrollierten Zugriff auf ihre Komponenten hat. Jede derartige ZE soll ein eigenes Schlüsselpaar besitzen, das von der ZE vor der Auslieferung selbst erzeugt wird, da der geheime Schlüssel

(SK_{ZE}) so niemals die ZE selbst verlassen muß und der Hersteller den öffentlichen Schlüssel (PK_{ZE}) direkt nach der Erzeugung zertifizieren kann.

Geheimnis der ZE	SK_{ZE}
ZE-Zertifikat	$\langle Id_{ZE}, PK_{ZE} \rangle_{ZEH}$

Der Schlüsselspeicher für den geheimen Schlüssel der ZE ist eine spezielle Hardwareerweiterung einer konventionellen ZE, der zu existierenden ZE nur hinzugefügt werden muß. Er wird so konstruiert, daß er nur nach einem Reset gelesen werden kann und vom Bootprogramm anschließend gesperrt wird. Dies garantiert, daß der geheime Schlüssel nicht von der Software ausspioniert werden kann. Sollte das Schlüsselpaar einer ZE von einem Angreifer doch kompromittiert werden, so könnte er die Identität dieser ZE vortäuschen. Das Ausspionieren des geheimen Schlüssels erfordert jedoch Manipulationen an der Hardware. Durch geeignete Konstruktion des Schlüsselspeichers führen solche Manipulationen zum Verlust des Schlüssels, was ein Ausspionieren unmöglich macht. Der Verlust des Schlüssels ist dann auch ein Hinweis auf eine Manipulation an der Zentraleinheit.

Jede vertrauenswürdige ZE soll zusätzlich eine Bootanzeige enthalten. Diese muß im Gegensatz zur Konsole in das Gehäuse integriert werden und darf nicht durch die Software beeinflußbar sein. Eine solche Anzeige kann deshalb dem Anwender unfälschbar das aktive BS anzeigen.

4 Ablauf

Das Vorgehen zur Erzeugung und Authentifizierung von Basissystemen läßt sich in drei Phasen unterteilen:

Phase 1 Vorbereitungsphase

In dieser Phase wird die ZE bei ihrem Hersteller initialisiert und der Security-Boot-Modul vom BS-Hersteller unterschrieben.

Phase 2 Vertrauenswürdiges Booten

Der Security-Boot-Modul wird geladen und authentifiziert. Die Bootanzeige wird gesetzt und er bekommt ein Schlüsselpaar und die notwendigen Zertifikate. Anschließend wird der BS-Lader gestartet.

Phase 3 Betriebsphase

Vom Start des BS-Laders bis zum Ausschalten oder dem nächsten Reset befindet sich das System in der Betriebsphase. Das BS wird hochgefahren und kontrolliert nun die ZE. Das entstandene Basissystem kann sich gegenüber dem Benutzer oder einem anderen Basissystem authentifizieren.

4.1 Phase 1 (Vorbereitungsphase)

Phase 1.a ZE-Initialisierung beim Hersteller.

a) Vor der ersten Inbetriebnahme schreibt der Hersteller in das EPROM der ZE ihre Id_{ZE} und den BP-Code.

Id_{ZE} und BP-Code

b) Die ZE erzeugt dann beim ersten Einschalten ein Schlüsselpaar (SK_{ZE}, PK_{ZE}) und gibt den öffentlichen Schlüssel aus. Der geheime Schlüssel (SK_{ZE}) wird im ZE-Schlüsselspeicher abgelegt.

PK_{ZE}

c) Anschließend erzeugt der Hersteller das ZE-Zertifikat, das er der ZE für spätere Authentifizierungen ihres öffentlichen Schlüssels übergibt.

$\langle Id_{ZE}, PK_{ZE} \rangle_{ZEH}$

Phase 1.b Der Security-Boot-Modul wird erzeugt.

Für sein BS erzeugt der BS-Hersteller eine Id_{BS} und signiert sie gemeinsam mit dem BS-Lader. Der entstandene Security-Boot-Modul wird zusammen mit den anderen Teilen des BS ausgeliefert.

$[Id_{BS}$, BS-Lader $]_{BSH}$

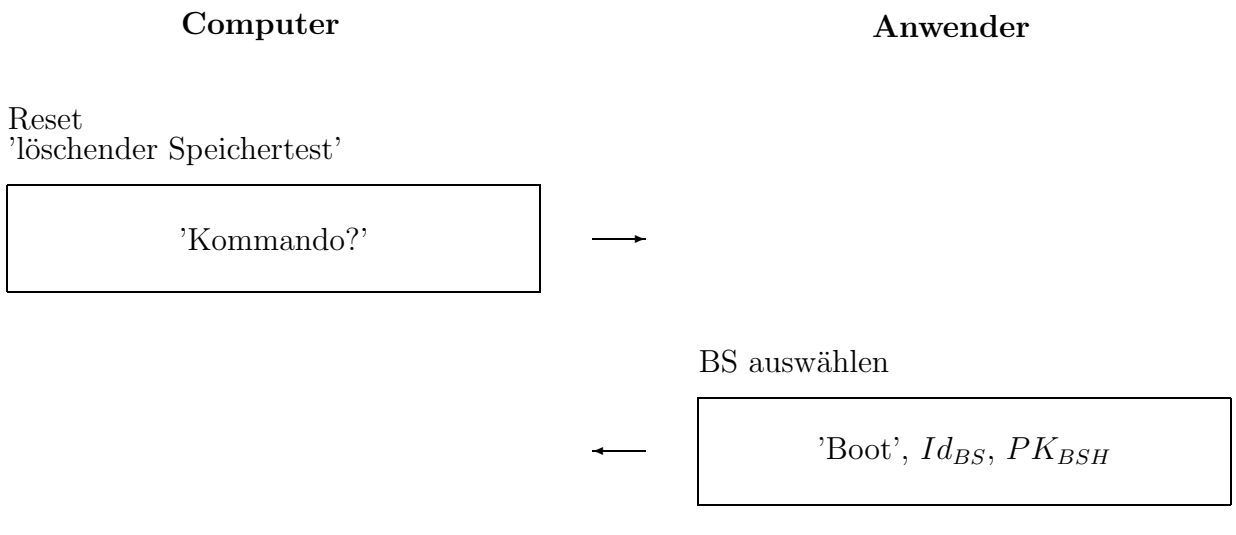
4.2 Phase 2 (Vertrauenswürdiges Booten)

Nach einem Reset der ZE wird zunächst das Bootprogramm gestartet, welches nach den üblichen Funktionsprüfungen der ZE und einem löschenden Speichertest den Bootvorgang durchführt. Die Zerstörung des alten Speicherinhalts garantiert dabei, daß Daten, die sich vor dem Reset im Hauptspeicher befanden, sicher gelöscht werden. Beim anschließenden vertrauenswürdigen Booten werden das BS und die ZE so zu einem Basissystem kombiniert, daß sich das Basissystem gegenüber anderen authentifizieren kann.

Nachdem der Security-Boot-Modul in den Hauptspeicher geladen wurde, wird die Signatur des BS-Herstellers geprüft. Dazu benötigt die Zentraleinheit den öffentlichen Schlüssel des BS-Herstellers. Anschließend erzeugt die Zentraleinheit das Bootzertifikat, welches die Identität des geladenen BS (Id_{BS}), den öffentlichen Schlüssel des erzeugten Schlüsselpaares (PK_{BS}) und den öffentlichen Schlüssel des BS-Herstellers (PK_{BSH}) enthält. Da ab jetzt der geheime Schlüssel der ZE nicht mehr benötigt wird, kann der ZE-Schlüsselspeicher nun gesperrt werden. Dies garantiert, daß nur das Bootprogramm der ZE Zugriff auf

den Schlüsselspeicher hat und niemand sonst eine Möglichkeit hat, den geheimen Schlüssel der ZE auszuspionieren. Als nächstes wird auf der Bootanzeige der Name des Betriebssystems angezeigt und die Bootanzeige wird ebenfalls gesperrt, damit kein Programm die Möglichkeit hat, diese Anzeige zu kompromittieren. Am Ende des Bootvorgangs wird der BS-Lader gestartet, der dann das komplette BS lädt, prüft und startet. Das laufende BS hat nun Zugriff auf seinen geheimen Schlüssel, das ZE-Zertifikat und das Bootzertifikat. Diese Dinge können vom BS benutzt werden, um sich anderen Basissystemen gegenüber zu authentifizieren oder abhörsichere authentische Kommunikationskanäle zu realisieren.

Für den Bootvorgang ergibt sich damit folgendes Szenario:



4.3 Phase 3 (Betriebsphase)

Zur Authentifizierung des Basissystems gibt es verschiedene Möglichkeiten:

a) Die Bootanzeige zur Authentifizierung des BS direkt durch den Anwender, unter der Voraussetzung einer nicht manipulierten, authentischen ZE.

b) Die Authentifizierung eines BS und einer ZE von einem anderen Basissystem aus, zur Realisierung eines sicheren verteilten Systems.

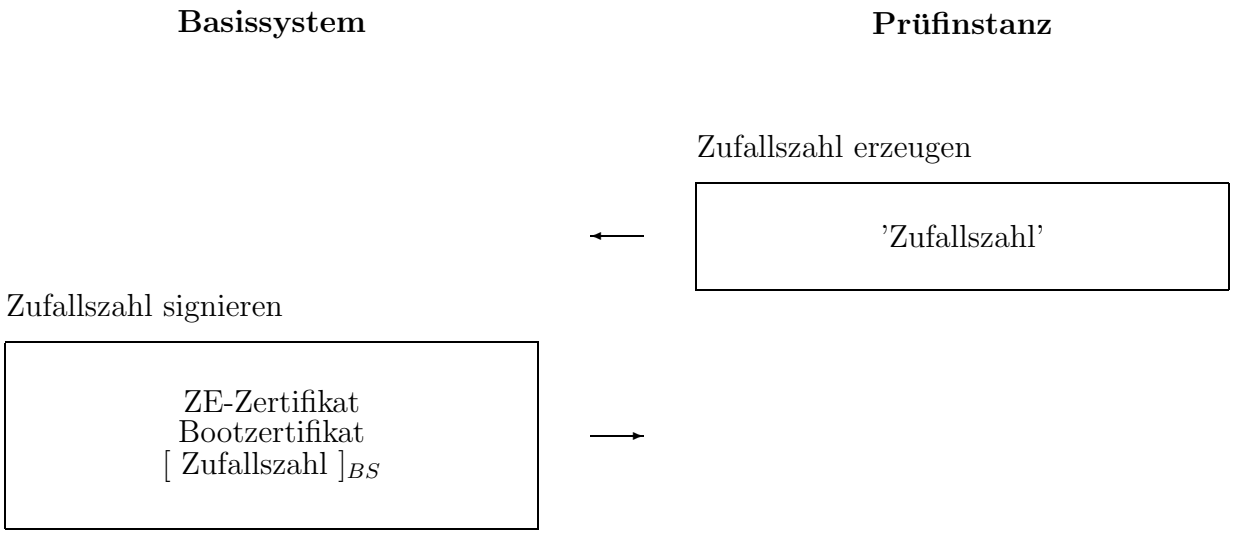
Die bereitgestellten Zertifikate und das Schlüsselpaar des BS können als Grundlage für die Durchsetzung verschiedener Sicherheitsmechanismen mit kryptographischen Verfahren dienen. Hier soll als Beispiel die Authentifizierung eines Basissystems gezeigt werden. Weitere Anwendungen werden in [Gro91] beschrieben.

Beispiel für die Authentifizierung eines Basissystems

Um ein Basissystem während des Betriebs zu authentifizieren, muß die Prüfinstanz die authentischen öffentlichen Schlüssel der Hersteller von ZE und BS kennen. Dies könnte zum Beispiel wie bereits oben beschrieben mit Hilfe von Zertifikaten einer Herstellervereinigung erreicht werden.

Zunächst überträgt man eine nie zuvor gesendete Nachricht (Nonce) an das zu prüfende Basissystem. Diese Nachricht wird vom BS mit seinem geheimen Schlüssel (SK_{BS}) unterschrieben und dann zusammen mit dem ZE-Zertifikat und dem Bootzertifikat zurückgeschickt. Die Eigenschaft der Nachricht, nie zuvor gesendet worden zu sein, ist notwendig, damit die signierte Nachricht nicht mehrmals verwendet werden kann. Sie kann z.B. durch eine Kombination von Datum und Zufallszahl erreicht werden. Als erstes prüft man die Signatur des Herstellers der ZE mit seinem bereits bekannten öffentlichen Schlüssel (PK_{ZEH}). Wenn diese Signatur authentisch ist, kennt man den authentischen öffentlichen Schlüssel der Zentraleinheit (PK_{ZE}), der zur Prüfung der Signatur des Bootzertifikats verwendet werden muß. Ist auch diese Signatur authentisch, so kennt man anschließend die Identität des BS (Id_{BS}) und des Herstellers (Id_{BSH}) sowie den öffentlichen Schlüssel (PK_{BSH}), der zum Prüfen des Security-Boot-Moduls verwendet worden war. Ein Vergleich mit dem bereits bekannten authentischen Schlüssel des BS-Herstellers zeigt dann die Authentizität des BS. Am Ende wird die Signatur des BS überprüft. Diese Signatur zeigt, daß das identifizierte BS auch wirklich die ZE kontrolliert.

In unserem Authentisierungsbeispiel ergibt sich damit folgendes Szenario:



5 Ausblick

Der vorgeschlagene Bootmechanismus ermöglicht die Konstruktion sicherer verteilter Systeme. Eine geplante Implementierung von vertrauenswürdigem Booten irn Rahmen des BirliX-Projektes soll die praktische Anwendbarkeit des Verfahrens zeigen. Im BirliX-System soll das Verfahren als Basis zur Durchsetzung der in [KH90] beschriebenen Si-

merheitsmechanismen dienen. Die im vorliegenden Papier nicht diskutierten Details und Alternativen werden demnächst in einem Arbeitsbericht der GMD 'Die Implementierung von vertrauenswürdigem Booten im BirliX-System' [Gro91] dargestellt werden. Im Kontext der DEC-Sicherheitsarchitektur wurde das Problem der Authentizität von Systemkomponenten ebenfalls erkannt und in [GGKL89] diskutiert. Durch die geplante Implementierung von vertrauenswürdigem Booten in Verbindung mit BirliX soll ein konkreter Weg zur Konstruktion verteilter Systeme, auf deren Sicherheit der Anwender vertrauen kann, aufgezeigt werden.

Anhang

Abkürzungen

E	Zentraleinheit
EH	Hersteller einer Zentraleinheit
S	Betriebssystem
SH	Hersteller eines Betriebssystems
HV	Herstellervereinigung

Notation

PK_X	öffentlicher Schlüssel von X
SK_X	geheimer Schlüssel von X
$[\ \text{Nachricht}\ ]_X$	Nachricht mit digitaler Signatur von X
$\langle\ Id_Y, PK_Y\ \rangle_X$	Zertifikat der Instanz X für die Instanz Y

Datenstrukturen

ZE-Geheimnis	SK_{ZE}
ZE-Zertifikat	$\langle Id_{ZE}, PK_{ZE} \rangle_{ZEH}$
Bootgeheimnis	SK_{BS}
Bootzertifikat	$\langle Id_{BS}, PK_{BS}, PK_{BSH} \rangle_{ZE}$
Security-Boot-Modul	$[\ Id_{BS}, BS - Lader]_{BSH}$
ZEH-Zertifikat	$\langle Id_{ZEH}, PK_{ZEH} \rangle_{HV}$
BSH-Zertifikat	$\langle Id_{BSH}, PK_{BSH} \rangle_{HV}$

Wo sollen die Daten gespeichert sein?

- externer Speicher (z.B. Platte, Diskette oder Band):

  Security-Boot-Modul

  Betriebssystem

- Arbeitsspeicher (RAM):

  Bootgeheimnis

  Bootzertifikat

- EPROM:

  Id_{ZE}

  BP-Code

- ZE-SchIüsseIspeicher:

  ZE-Zertifikat

  ZE-Geheimnis

Literatur

[Cle88] Wolfgang Clesle. Schutz auch vor Herstellern und Betreibern von Informationssystemen. Master's thesis, Universität Karlsruhe, Institut für Rechnerentwurf und Fehlertoleranz, Prof. W. Görke, Juni 1988.

[Den82] Dorothy E. Denning. *Cryptography and Data Security.* Addison-Wesley, Reading, Massachusetts, 1982.

[DH76] Whitfield Dime and Martin E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, pages 22(6):644–654, November 1976.

[DP84] D. W. Davies and W. L. Price. *Security for Computer Networks.* John Wiley & Sons, Chichester, 1984.

[EHV88] J. Ekberg, S. Herda, and J. Virtamo. TeleTrust - Technical Concepts and Basic Mechanisms. In *Proceedings of EUTECO '88*, pages 523–533, 1988.

[GGKL89] Morrie Gasser, Andy Goldstein, Charlie Kaufman, and Butler Lampson. The Digital Distributed System Security Architecture. In *Proceedings of 1989 National Computer Security Conference*, 1989.

[Gro91] Michael Groß. Die Implementierung von vertrauenswürdigem Booten im BirliX-System. Arbeitspapiere der GMD, in Vorbereitung, 1991.

[KH90] 0. C. Kowalski and H. Härtig. Protection in the BirliX Operating System. In *Proceedings of the 10th International Conference on Distributed Computing Systems*, pages 160–166. IEEE, May 1990.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. In *Communications of the ACM*, pages 21(2):120–126, 1978.

[Zim86] P. Zimmermann. A Proposal Standard Format for RSA Cryptosystems. In *Computer*, pages 19(9):21–34, September 1986.