Darstellung und Analyse eines Konzeptes zur digitalen Beweissicherung

Bachelor-Arbeit

vorgelegt von	Kai T. Hillmann khillman@tzi.de
Matrikelnummer	22 111 80
betreut durch	Dr. Karsten Sohr Prof. Dr. Michael Lawo
vorgelegt am	13. Oktober 2011

Universität Bremen

Erklärung

Ich versichere, den Bachelor-Report oder den von mir zu verantwortenden Teil einer Gruppenarbeit*) ohne fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen sind, sind als solche kenntlich gemacht.

*) Bei einer Gruppenarbeit muss die individuelle Leistung deutlich abgrenzbar und bewertbar sein und jeweils den Anforderungen aus § 12 Abs. 1 der Prüfungsordnung für den Studiengang Bachelor of Science Informatik entsprechen.

Bremen, den 13. Oktober 2011

_____

(Unterschrift)

Inhalt

Kapitel 1

Einleitung

1.1 Motivation

In der heutigen Zeit werden immer mehr geschäftliche und private Situationen bzw. Ereignisse in digitaler Form erfasst und archiviert.

In einem Unternehmen werden z. B. Meetings in Protokollen festgehalten, Mailkommunikation wird teilweise über Jahre hinweg archiviert (vgl. [BMJ11a, §147]) und selbst das Betreten oder Verlassen der Geschäftsräume wird ggf. erfasst.

Im privaten Umfeld sammeln, bedingt durch die niedrigen Preise für Speichermedien, nicht wenige inzwischen ein oder mehrere Terabyte an Daten. Angefangen von Mails über Multimediainhalte bis hin zu Bank-Transaktions- oder Zugangsdaten.

Auch wenn heutzutage beinahe alles digitalisiert wird, werden digitale Informationen aber vor Gericht nicht immer als Beweismittel verwendet oder zugelassen, da die Gerichte hohe Ansprüche an die Glaubwürdigkeit der Beweismittel haben (vgl. [Fis+02, S.709], [Wohoo, S.8] und [Scho9]). Dabei wären oftmals genügend digitale Informationen vorhanden, da es kaum etwas gibt, was nicht digital gespeichert wird.

Das Problem bei der Verwendung digitaler Informationen als Beweis liegt dabei im eigentlichen Vorteil dieser digitalen Daten, der einfachen Veränderung bzw. Kopie. Durch die digitale Form und die damit einhergehende leichte Veränderbarkeit genügen diese meist nicht den hohen Ansprüchen der Justiz hinsichtlich der Glaubwürdigkeit von Beweisen. Die Justiz steht damit vor dem Dilemma, dass immer mehr Daten ausschließlich in digitaler Form vorliegen, die Glaubwürdigkeit dieser aber oftmals nicht mal von Experten eindeutig erklärt oder eingeschätzt werden kann.

Da es in der Zukunft aber immer öfter dazu kommen dürfte, dass weniger herkömmliche Beweise zu finden sind, wird auf diesem Feld aktiv geforscht um sicherstellen zu können, dass die digitale Dokumentation realer Ereignisse eindeutig und unwiderlegbar ist. Um das bewerkstelligen zu können muss die Nichtabstreitbarkeit bzw. Verbindlichkeit (vgl. [Ecko9, S. 11,736,742]) der Daten gewährleistet sein.

Ein Konzept, das versucht, die Sicherung der digitalen Beweise hinsichtlich der Nichtabstreitbarkeit und unter Zuhilfenahme von Konzepten und Methoden des Trusted Com-

puting zu lösen, wurde dabei in der Diplomarbeit "Digital Evidence"[Ric09] in Zusammenarbeit mit dem deutschen Fraunhofer Institut für Sicherheit in der Informationstechnik in Darmstadt erarbeitet. In dieser Arbeit und einem zusammenfassenden Artikel [RKR10] wird als Anwendungsbeispiel ein digitales Messgerät zur Verkehrsüberwachung z. B. Geschwindigkeits- oder Abstandsüberwachung betrachtet. Mögliche weitere Einsatzgebiete wären dabei z.B. folgende :

- Zeiterfassungssysteme in Unternehmen
  Systeme zur Erfassung der geleisteten Arbeitszeit erfassen beispielsweise den Beginn einer Arbeitstätigkeit, indem der Arbeitnehmer mit einer RFID-Chipkarte an einem Erfassungsgerät vorbei läuft, eine Smartcard in ein Lesegerät an einem Zugangsdrehkreuz einführt oder schlicht wenn dieser sich an seinem Computer anmeldet. Danach wartet das System auf ein Ende der Arbeitstätigkeit und erfasst diese z. B. auf dem gleichen Wege wie den Beginn. Die ermittelten Arbeitszeiten werden dabei i. d. R. für Abrechnungs- und Nachweiszwecke verwendet, könnten jedoch potentiell auch ein Alibi (vgl. [WS10, Kapitel 13 Abschnitt 6]) sein. Weiterhin ist denkbar, dass bei Angriffen auf das interne Unternehmensnetz so die Anwesenheit der Mitarbeiter im Unternehmen nachgewiesen werden kann.
- SmartHome Lösungen
  Der Begriff SmartHome (vgl. [JLY04]) bezeichnet eine Steuerung und oder Erfassung (vgl. Smartmetering [BHJP09]) der Nutzung der Grundressourcen Wasser, Strom und Gas in einem Haushalt. Ähnliche Systeme sind aber auch im betrieblichen Umfeld denkbar. Dabei zielen die Versorgerkonzerne prinzipiell darauf, ihre Ressourcen durch genauere Informationen, wann diese benötigt werden, besser und kostengünstiger bereitstellen zu können. Für den Verbraucher soll ein Mehrwert durch eine flexiblere Steuerungsmöglichkeit der hausinternen Verbraucher sowie ggf. günstigere von der aktuellen Gesamtnutzung abhängige Tarife entstehen. [BHJP09]
- Clientmanagement Paket-/Updateinstallation
  Bei der Administration großer IT-Systeme fallen einige standardmäßige Service-Aufgaben an. Diese könnten z. B. sein, Softwarepakete und oder -updates auf eine größere Anzahl von Endsystemen zu verteilen oder einen Status der derzeitig

installierten Softwarepakete der erfassten Systeme zu ermitteln. Dabei könnten sicherheitskritische Umgebungen die Anforderungen haben, dass nachweisbar sein muss, zu welchem Zeitpunkt welche Updates installiert oder zur Installation angeboten und dann ggf. vom Benutzer installiert oder ignoriert worden sind. Dies kann im speziellen dann relevant werden, wenn es um die Verantwortlichkeiten für Fehlverhalten der Endsysteme geht.

- Mails / Instantmessaging
  Bei Mails und Instantmessaging könnte insbesondere der Aspekt interessant sein, wie eindeutig gemacht werden kann, dass eine Nachricht beim Empfänger (dem Benutzer) angekommen ist. Auch bei dem Versand von Nachrichten muss gewährleistet sein, dass der Benutzer, der diese Nachrichten geschrieben hat eindeutig identifizierbar ist. Außerdem muss sichergestellt werden, dass der Autor der Nachricht diese bewusst geschrieben und abgeschickt hat und die Nachricht nicht ohne seine Mitwirkung durch Schadprogramme oder ein kompromittiertes System versendet wurde.
- Filesharing
  Ebenso wie bei Mails / Instantmessaging muss auch für einen sicheren Austausch von Dateien sichergestellt sein, dass die ausgetauschten Daten tatsächlich vom Empfänger stammen, unverändert sind und ggf. anschließend unwiderlegbar ist, welche Dateien zu welchem Zeitpunkt von wem ausgetauscht worden sind. Auch hier muss eindeutig belegt werden, dass die Übertragung der Dateien durch den Willen des Benutzers und nicht durch manipulierte Programme ohne dessen Kenntnis durchgeführt wurde.

1.2 Kontext

Der resultierende Bachelorreport beschäftigt sich, wie bereits genannt, mit einem existierendem Konzept zur Realisierung der geeigneten automatisierten Erfassung und Verarbeitung von Informationen zur Verwendung vor Gericht.

Im Zuge dessen bestehen einige Annahmen zur erwarteten Umgebung, in der ein Trusted Platform Module (TPM) vorhanden und von dem angenommen wird, dass dieses sicher ist. Des Weiteren wird angenommen, dass die genutzten und in der Literatur bekannten kryptografischen Algorithmen oder Verfahren sicher sind. Außerdem gibt es einige Richtlinien bzw. Gesetze, die eine gesetzliche Grundlage rund um digitale Beweise definieren.

Dadurch, dass die erwarteten Anwendungszwecke i. d. R. eher von leistungsarmen und günstigen Systemen umgesetzt werden dürften, spielt darüber hinaus der Aspekt der Performanz eines möglichen Konzeptes eine größere Rolle.

Diese und weitere Annahmen werden in Abschnitt 3.1.2 (Seite 48) ausführlicher beschrieben.

1.3 Aufgabenstellung

Wie bereits in Abschnitt 1.1 (Seite 2) erwähnt, soll sich diese Arbeit mit der übergeordneten Fragestellung, wie sich digital ermittelte Daten beweiskräftig erheben lassen beschäftigen. Dabei sollte das vorhandene Konzept ursprünglich in eine saubere, modulare und wiederverwendbare bzw. erweiterbare Implementierung überführt werden. Im Verlauf der Analyse, der zu Grunde liegenden Arbeit [Ric09], hat sich jedoch herausgestellt, dass, um die konzeptuellen Zusammenhänge der Arbeit und vor allem einiger kryptographischer Konzepte des Trusted Computing zu verstehen, weitere Informationen und Konzepte recherchiert werden mussten. Zudem gab es in einigen Bereichen Unklarheiten, durch widersprüchliche Informationen.

Durch diese Erkenntnisse wurde das ehemalige Ziel, eine Implementierung des Konzeptes zu erstellen und diese zu diskutieren, nahezu unmöglich gemacht. Nachdem bei der Betrachtung einiger Kernpunkte des Konzeptes ein vermeintlicher Fehler gefunden wurde, ist das Ziel der Arbeit dahingehend geändert worden, das Konzept möglichst weit abstrahiert bzw. vereinfacht und Kernbereiche ggf. auch formalisiert darzustellen.

Die Darstellung soll dabei einen möglichst hohen Standardisierungsgrad erreichen, um später leichter von unabhängigen Experten analysiert und diskutiert werden zu können. Auch sollen exemplarisch Kernkomponenten kritisch betrachtet und ggf. gefundene Fehler oder nicht eindeutige Abläufe aufgezeigt werden.

Ferner soll durch die auf diesem Wege entstandene Arbeit eine zukünftige saubere Implementierung oder formale Verifikation erleichtert werden.

1.4 Vorgehensweise

In der vorliegenden Arbeit wird zunächst im Kapitel 2 (Seite 8) auf die Vorkenntnisse eingegangen, die benötigt werden um das Konzept [Ric09] verstehen zu können. Dabei werden neben den Schutzzielen (vgl. [Eck09, S.6-14], [And08, S.11-15], [BPS10]) auch grundlegene kryptographische Verfahren erklärt und schließlich eine Einführung in benötigte Konzepte des Trusted Computing, in Sicherheitsprotokolle und in eine bekannte formale Notationsform gegeben.

Um das Konzept richtig einordnen zu können, wird danach außerdem kurz der rechtliche Rahmen beschrieben, in dem sich die Verfahren bewegen müssen. In Kapitel 3 (Seite 44) wird dann das vorliegende Konzept [Ric09] dargestellt und zusammengefasst bzw. vereinfacht. Am Ende des Kapitels wird in Abschnitt 3.2 (Seite 59) ein Kernbestandteil des Konzeptes analysiert, um exemplarisch einige Probleme bzw. Unklarheiten aufzuzeigen.

In Kapitel 4 (Seite 66) wird eine alternative Form der Darstellung eingeführt bzw. vorgeschlagen und anhand von Beispielen erläutert.

Schlussendlich legt Kapitel 5 (Seite 74) die vom Autor gewonnenen Eindrücke über die Komplexität des *Trusted Computings* und die im Konzept verwendeten Darstellungsformen dar. Zudem werden weitere Verwendungsmöglichkeiten der vorgeschlagenen Notation diskutiert.

Kapitel 2

Grundlagen

In diesem Kapitel werden im Folgenden einige Grundlagen erläutert, die zum Verständnis des Konzeptes [Ric09] bzw. zu den in Abschnitt 1.4 (Seite 6) beschriebenen weiteren Kapiteln nötig sind.

2.1 Schutzziele

Bei der Verarbeitung digitaler Informationen gibt es verschiedene Anforderungen hinsichtlich der Sicherheit der Daten. Dabei wird unterschieden zwischen der Sicherung der Daten gegenüber Angriffen und der Sicherung der Daten gegenüber Fehlfunktionen bzw. fehlerhaftem Verhalten [BPS10, Foliensatz 0, Folie 17].

Die bestehenden Anforderungen werden dabei vereinheitlicht durch so genannte Sicherheitsziele bzw. Schutzziele beschrieben. Dies dient unter anderem dazu, bei der Ermittlung bzw. Diskussion des Schutzbedarfes eines IT-Systems über einheitlich definierte Problemstellungen sprechen zu können.

Bei einer Beschreibung eines möglichen Angriffs auf ein IT-System würden sonst u. U. verschiedene Sicherheitsziele vermischt, und es ließe sich nicht ohne größeren Aufwand unterscheiden, ob zwei verschiedene genannte Angriffsrisiken nicht prinzipiell die gleichen Anforderungen haben. In diesem Abschnitt sollen daher vor allem die für diese Arbeit relevanten Sicherheitsziele kurz aufgelistet und erklärt werden.

Wie bereits in Abschnitt 1.1 (Seite 2) erwähnt, ist eines der wichtigsten Sicherheitsziele die „Nichtabstreitbarkeit“ bzw. „Verbindlichkeit“. Hinzu kommt jedoch, gerade bei dem Beispiel des Verkehrsüberwachungsmessgerätes aus dem Konzept [Ric09], das Sicherheitsziel „Datenschutz“, da ein Autofahrer zwar für eine Übertretung des Geschwindigkeitslimits zur Rechenschaft gezogen werden muss, die Information, dass oder wo dieser die Geschwindigkeit in welchem Maße übertreten hat, im Zweifel jedoch nicht Unbeteiligten preisgegeben werden darf. Des Weiteren bedingt sich aus der „Verbindlichkeit“ die „Integrität“, „Authentizität“ und „Zurechenbarkeit“.

Im Folgenden werden nun die für diese Arbeit wichtigsten Sicherheitsziele erklärt. Dabei ist, wenn von einer Identität gesprochen wird, gleichermaßen ein IT-System, eine einzelne Software-Komponente oder eine natürliche Person gemeint.

- „Integrität“
  Wenn von dem Sicherheitsziel der Integrität gesprochen wird, geht es dabei vor allem darum, dass die Daten, auf die dieses bezogen ist, in exakt unveränderter Form vorliegen oder eine Veränderung auf jeden Fall entdeckt wird bzw. nachweisbar ist.
- „Authentizität“
  Steht für den Schutz der „Integrität“ und Herkunft von Daten, um diese eindeutig Identitäten zuordnen zu können.
- „Zurechenbarkeit“
  Die Zurechenbarkeit soll sicherstellen, dass eine Menge von authentischen Daten einer Handlung und einer Identität eindeutig zugeordnet werden kann.
- „Verbindlichkeit/Nichtabstreitbarkeit“
  Bei der Verbindlichkeit soll sichergestellt werden, dass die zu einer Handlung gehörenden Daten einer Identität zuzuordnen sind und von der zu Grunde liegenden Identität die gesamte durchgeführte Handlung gegenüber einer unabhängigen und neutralen dritten Instanz, beispielsweise eines Richters, nicht abgestritten werden kann.
- „Datenschutz“
  Beim Datenschutz soll das Grundrecht auf informationelle Selbstbestimmung eines jeden Menschen in dem Sinne geschützt werden, als dass personenbezogene Daten nur aufgrund gesetzlicher Bestimmungen oder mit Zustimmung der Person, auf die sich die Daten beziehen, von einem IT-System verarbeitet oder preisgegeben werden dürfen. Das Schutzziel des Datenschutzes leitet sich vom Schutzziel der Vertraulichkeit ab und wird von Pfitzmann in [PH10] noch weiter untergliedert.

2.2 Angriffsformen

Die im vorhergehenden Abschnitt 2.1 beschriebenen Schutzziele werden i. d. R. von einer ganzen Reihe von typischen Angriffen gefährdet. In diesem Abschnitt werden daher zwei häufige Angriffstypen, die auch in der Beschreibung des Konzeptes von Richter [Ric09] verwendet werden, beschrieben. Bei den im Konzept erwähnten Angriffstypen handelt es sich unter anderem um einen so genannten *Man-In-The-Middle (MiM)*-Angriff und einen *Denial-of-Service (DoS)* bzw. *Distributed-Denial-of-Service (DDoS)*. Bei einem *Man-In-The-Middle (MiM)*-Angriff wird die Authentizität, Vertraulichkeit bzw. der Datenschutz eines Kommunikationspartners angegriffen. Ein Angreifer platziert sich bei einer Kommunikation zwischen zwei Parteien A und B logisch zwischen diesen. Um dies zu erreichen, gibt er sich gegenüber A als B und gegenüber B als A aus und leitet die Nachrichten von A bzw. B weiter. *Man-In-The-Middle*-Angriffe nutzen i. d. R. fehlerhafte Authentifizierungsprotokolle oder sonstige Sicherheitsprotokollfehler aus. Ein *Denial-of-Service (DOS)*-Angriff hingegen zielt darauf ab, die Verfügbarkeit eines Systems einzuschränken, um daraus einen Vorteil zu erzielen. Dies wird versucht durch Überflutung bzw. Überlastung des Zielsystems mit Nachrichten bzw. manipulierten Nachrichten zu erreichen und ggf. dadurch bedingte Abstürze des Systems. Von einem *Distributed-Denial-of-Service (DDoS)*-Angriff wird gesprochen, wenn die „Bombardierung“ des Zielsystems nicht nur durch einen einzelnen Rechner eines Angreifers, sondern durch gezielte verteilte Systeme durchgeführt wird.

2.3 Kryptographie

Um die in Abschnitt 2.1 (Seite 8) genannten Schutzziele zu erreichen, wird eine Lösung benötigt, um Schlüssel- und andere Informationen unter Erhalt der Vertraulichkeit austauschen zu können. Die Kryptographie liefert hierfür einige Ansatzpunkte. Dazu baut die Kryptographie auf der Mathematik auf. Sie bedient sich dieser, um gemäß dem dortigen aktuellen Kenntnisstand es möglich zu machen, Klartextinformationen in eine ver-

schlüsselte Form zu bringen und den dabei erreichten Schutzgrad der Informationen auf Basis der mathematischen Komplexität beurteilen und nachweisen zu können. Es wird in der Kryptographie unterschieden zwischen symmetrischen und asymmetrischen Verfahren.

2.3.1 Symmetrische Kryptographie

Symmetrische Verfahren basieren grundsätzlich darauf, dass zwei oder mehrere Kommunikationspartner ein gemeinsames Geheimnis haben, welches im Vorfeld über einen sicheren Kanal ausgetauscht wurde. Von diesem Geheimnis werden dann entweder weitere Schlüssel abgeleitet bzw. erzeugt oder dieser wird für einzelne Handlungen direkt benutzt.

2.3.2 Asymmetrische Kryptographie

Die asymmetrischen Verfahren hingegen benötigen kein gemeinsames Geheimnis bei allen Kommunikationspartnern, basieren jedoch oft auf der Primfaktorzerlegung (vgl. [RSA82]) und in verstärktem Maße auf Vertrauen. In diesen wird mit einem öffentlichen und einem privaten Schlüssel gearbeitet. Informationen werden zur Übertragung mit dem öffentlichen Schlüssel des Kommunikationspartners verschlüsselt, da nur dieser, gemäß der mathematischen Grundlage, in der Lage ist, die Informationen mit seinem privaten Schlüssel korrekt zu entschlüsseln. Die Herausforderung hier liegt aber zum einen in der Verwendung entsprechend großer Schlüssellängen, um die mathematische Komplexität aufrecht zu erhalten, zum anderen in der eindeutigen Identifikation bzw. Authentifizierung des korrekten öffentlichen Schlüssels des gewünschten Kommunikationspartners.

Durch die größeren verwendeten Schlüssellängen und der häufig verwendeten aufwändigen Primfaktorzerlegung ist die asymmetrische Kryptographie für eingebettete Systeme oftmals weniger geeignet, da diese entsprechende Rechenkapazitäten voraussetzt.

2.3.3 Signatur und X.509-Zertifikate

Das bei asymmetrischer Kryptographie auftauchende Problem der Integrität und Authentizität von öffentlichen Schlüsseln der Kommunikationspartner wird dabei durch die Ausstellung von Zertifikaten adressiert.

Dabei wird von einer vertrauenswürdigen Zertifizierungsstelle, deren privater Schlüssel physisch besonders gesichert ist, der öffentliche Schlüssel einer Identität nach Prüfung dieser signiert und diese Signatur zusammen mit allen relevanten Daten in einem Zertifikat hinterlegt, welches dieser Identität ausgestellt und übermittelt wird. Die relevanten Daten sind hierbei z. B. der Name der Identität, der öffentliche Schlüssel der Identität, das Datum der Signierung, die Gültigkeitsdauer des Zertifikates, der Name der Zertifizierungsstelle, die Signatur der Zertifizierungsstelle und die verwendeten Algorithmen.

Dadurch, dass bei der Signatur mit dem privaten Schlüssel ein Hashwert des öffentlichen Schlüssels des Kommunikationspartners verschlüsselt wird, ist jeder, der den öffentlichen und i. d. R. frei zugänglichen Schlüssel der vertrauenswürdigen Zertifizierungsstelle besitzt, in der Lage, mit diesem den originalen Hashwert des beglaubigten öffentlichen Schlüssels des Kommunikationspartners zu entschlüsseln. Danach kann der so gewonnene Hashwert gegen einen eigenen, selbst über den für den Kommunikationspartner vorliegenden öffentlichen Schlüssel erzeugten, verglichen und somit die Glaubwürdigkeit überprüft werden.

Dieses Vorgehen wird dabei ebenso verwendet, um eine hierarchische Vertrauenskette zu bilden, auf dessen oberster Ebene so genannte Root-Zertifizierungsstellen stehen, deren Vertrauenswürdigkeit die Hersteller von IT-Systemen voraussetzen und die die Identität von kleineren Zertifizierungsstellen ihrerseits mittels Signatur und Zertifikat beglaubigen. Möglicherweise muss also bei der Prüfung der Identität eines Kommunikationspartners die ganze Kette sukzessive durchlaufen werden, um schlussendlich sicher zu sein, dass diese glaubwürdig ist.

2.3.4 Herausforderungen der Kryptographie

Wie in dem Abschnitt 2.3 (Seite 10) bereits beschrieben, basiert die kryptografische Sicherung von Kommunikationsvorgängen auf Schlüsselinformationen. Diese müssen die Kommunikationspartner geheim vorhalten, um auszutauschende Nachrichten zu ver- oder empfangene Nachrichten zu entschlüsseln. In der sicheren Speicherung und Geheimhaltung der Schlüssel liegt jedoch eine der größten Herausforderungen der Kryptographie.

Dies stellt eine Herausforderung dar, da es viele mögliche Wege gibt, an die vermeintlich geheimen Informationen zu gelangen. (vgl. [BPS10, Foliensatz 0, Folie 28]) So erleichtert es Angreifern beispielsweise erheblich die Arbeit, wenn nicht sichergestellt ist, dass ein System nicht verändert wird, Sicherheitslücken in den verwendeten Implementierungen behoben werden und menschliche Nutzer Schlüssel in analoger Form nicht unsicher aufbewahren.

Mit diesen ist es dann z. B. möglich wie in Abschnitt 2.3.3 (Seite 12) beschrieben, sich mit der Identität, deren Schlüssel erlangt wurde, auszuweisen bzw. mit dieser zu handeln. Auch ist damit eine Entschlüsselung der vormals geheimen Kommunikation dieser mit einer anderen möglich, sofern die verschlüsselte Kommunikation mitgeschnitten wurde.

2.4 Trusted Computing

Da die Arbeit bzw. das Konzept von Richter [Ric09] auf *Trusted Computing* aufbaut, werden hier in den folgenden Abschnitten die für das Verständnis des Konzeptes notwendigen Grundlagen erklärt. Dabei bietet Abschnitt 2.4.1 (Seite 14) zunächst einen groben Überblick über Gründe für das *Trusted Computing*. Nachfolgend wird von Abschnitt 2.4.2 (Seite 15) ein Überblick über die grundlegene Trusted Computing Group (TCG)-Trusted Platform (TP)-Architektur gegeben. In Abschnitt 2.4.3 (Seite 16) werden anschließend die Kernziele und ihre Realisierung beschrieben. Danach folgt die Beschrei-

bung der TPM-Architektur und -Fähigkeiten in Abschnitt 2.4.4 (Seite 24). Schlussendlich wird in Abschnitt 2.4.5 (Seite 27) die Softwareunterstützung für Anwendungsprogramme beschrieben, um die bis hierhin gesammelten Grundlagen dann in den zuletzt beschriebenen *Trusted Computing*-Konzepten in Abschnitt 2.4.6 (Seite 29) zu nutzen. Aufgrund der teilweise enormen Komplexität des Themengebiets des *Trusted Computings* werden in den folgenden Abschnitten häufig Abkürzungen verwendet. Um daher manche Teile dieser Arbeit leichter verstehen zu können, bietet es sich an, das im Anhang befindliche Abkürzungsverzeichnis als Referenz zu verwenden. Des Weiteren kommt es aufgrund der vielfältigen Abhängigkeiten innerhalb des Themengebietes zu Referenzen zu späteren Abschnitten, auf die hier aus Gründen der Nachvollziehbarkeit bewusst nicht verzichtet wurde.

2.4.1 Gründe

Die Konzepte, die die Grundlage für die Industriestandards des *Trusted Computings* bilden, wurden von der TCG [TCG11b] ausgearbeitet und spezifiziert, um unter anderem für die in Abschnitt 2.3.4 (Seite 13) angedeuteten Herausforderungen Lösungsansätze zu bieten, in denen bestimmte Teile der Sicherheitsinfrastruktur in Hardware statt in Software bereitgestellt werden. Ursprünglich sollten Systeme ermöglicht werden, die ihre Konfiguration gegenüber Dienste- bzw. Medienanbietern als zulässig attestieren können. Dies sollte es diesen ermöglichen, ihre urheberrechtlich geschützten Inhalte nur auf Systemen, die eine vorgegebene Konfiguration aufweisen, zugänglich zu machen. Ob die Nutzung für Digital Rights Management (DRM)-Systeme jedoch tatsächlich die hauptsächliche Motivation für die Entwicklung von *Trusted Computing*-Konzepten bzw. -Standards war oder ist, wird bezweifelt, kann aber auch nicht nachhaltig widerlegt werden (vgl.[Ecko9, S.588],[Safo2, S.5,6],[Ando4, S.3,4],[Ando3]).

Einen der wichtigsten Gründe für ihre Entwicklung nennt Dr. Roger R. Schell jedoch bereits 1973:

> However, from a practical standpoint, the security problem will remain as long as manufacturers remain committed to current system architecture,

> produced without a firm requirement for security. As long as there is support for ad hoc fixes and security packages for these inadequate designs and as long as the illusory results of penetration teams are accepted as demonstrations of a computer system security, proper security will not be a reality [RJP73, S.I4-I5].

Dabei wird vor allem der Aspekt berücksichtigt, dass bei der Konzeption vieler Softwaresysteme Sicherheitsaspekte nicht von vornherein eine große Rolle spielen, sondern die Hersteller versuchen nach Bekanntwerden von Sicherheitslücken, diese durch Updates zu schließen. Von Challenger et. al wird außerdem die Frage ob Softwaresysteme absolut sicher gemacht werden können verneint, da die Komplexität großer Systeme mit Milliarden von Quellcodezeilen zu hoch ist.(vgl. [Cha+07, S. 9]) Dabei haben laut Challenger et. al Studien gezeigt, dass während der Lebensdauer einer Softwareanwendung schätzungsweise ein Sicherheitsfehler auf etwa tausend Zeilen kommt. (vgl. [Cha+07, S.9]) Mit einem solchen Multiplikationsfaktor ist leicht nachvollziehbar, weshalb Software allein nicht in der Lage ist, absolute Sicherheit zu bieten.

Auch in [Ric09, S.14] wird aufgrund der zunehmenden Angreifbarkeit von Softwaresystemen eine Realisierung bestimmter sicherheitsrelevanter Funktionalitäten in Hardware bevorzugt.

2.4.2 Trusted Platform Architektur

Eine TP besteht i. d. R. aus einem hardwareseitigen TPM, einer entsprechenden Unterstützung dessen im Basic Input Output System (BIOS), im Bootloader sowie im Betriebssystem und einem so genannten TCG Software Stack (TSS), um aufsetzend auf das Betriebssystem, Anwendungsprogrammen die Möglichkeit zu geben, die Sicherheitsfunktionen des TPM zu nutzen. (vgl. Abbildung 2.1 (Seite 17), [TCG07, Kapitel 4])

Auch ein TP-System bietet jedoch nicht absolute Sicherheit, da auch die Hardware-basierten TPMs physisch oder durch zu geringe Schlüssellängen angreifbar sind (vgl. [Ric09, S.32-

35]) oder aufgrund der Systemkomplexität nicht alle ggf. erforderlichen Systemwerte in Konfigurationsberechnungen einbezogen werden können (vgl. [Eck09, S.615]).

2.4.3 Trusted Platform Kernfunktionen

Eines der primären Ziele, die das Design der TP-Spezifikationen erfüllen soll, ist das Bereitstellen von sicheren Funktionen, um auf speziell gesicherte Bereiche, auf die auf anderem Wege kein Zugriff besteht, z. B. im Random Access Memory (RAM) oder in speziellen Registern des TPM, zuzugreifen [TCG07, S.5].

Die geschützten Bereiche dienen dabei der sicheren Erfassung einer Systemkonfiguration, der sicheren Speicherung und Erzeugung von kryptografischen Schlüsseln für asymmetrische kryptographische Verfahren (vgl. Abschnitt 2.3.2 (Seite 11)), der sicheren Signierung von Daten und der Möglichkeit, die Konfiguration des Systems einem Dritten gegenüber zu attestieren (wird in Abschnitt 2.4.6.4 (Seite 34) beschrieben) oder einer Integrationsprüfung zu unterziehen. Da so die Systemkonfiguration sicher ermittelt werden kann, können die gespeicherten Schlüssel vor einem Zugriff durch modifizierte Software geschützt werden, da sich ihre Nutzung nur bei einer vorgegebenen Konfiguration ermöglichen (vgl. [Cha+07, S.10]) lässt.

Der wesentliche Gedanke der TCG [TCG11b] beruht auf einer transitiven Vertrauensbildung von einem Vertrauensanker hin zu einer Vertrauenskette. (vgl. auch [Mül08, S.27]) Den Grundstein dafür legt der so genannte Trusted Building Block (TBB). Die TCG definiert, dass dieser alle Komponenten enthält, die Teil des grundlegenden Vertrauensankers sind, aber nicht selbst über einen geschützten Speicher verfügen. Ein beispielhafter TBB gemäß TCG wird in Abbildung 2.1 (Seite 17) dargestellt.

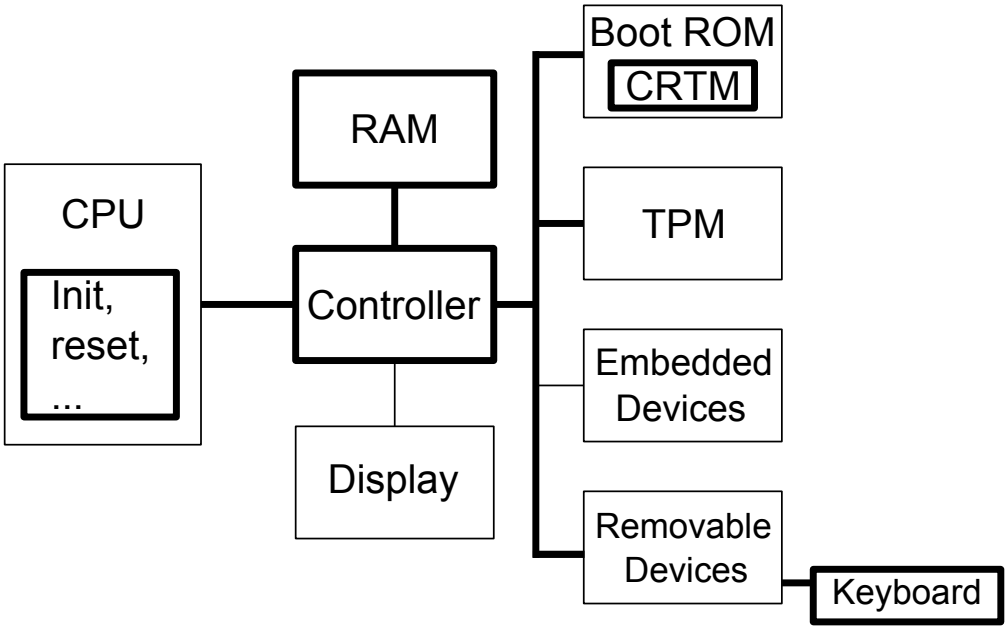

Abbildung 2.1 Die Fett-umrandeten Blöcke sind Teil des TBB.[TCG07, Abb.4b]

Für alle im TBB enthaltenen Komponenten gilt, dass angenommen wird, dass diese nicht kompromittiert und somit sicher sind. Diese Annahme stützt sich auf eine externe Überprüfung dieser Komponenten durch den Hersteller (vgl. [TCG07, S.6]), der durch Plattform-Zertifikate bestätigt, dass diese gemäß der Spezifikation funktionieren. Dies heißt im Umkehrschluss, dass den Komponenten des TBBs kompromisslos vertraut wird und eine Kompromittierung dieser unweigerlich den Verlust der Vertrauenswürdigkeit des Gesamtsystems zur Folge hat.

Der TBB besteht nach Müller aus den Vertrauensankern Root of Trust for Measurement (RTM), Root of Trust for Storage (RTS) und Root of Trust for Reporting (RTR).(vgl. [Mülo8, S.27]) Der RTM beinhaltet dabei den so genannten Core Root of Trust for Measurement (CRTM), welcher die Basis für die sichere Erfassung von Integritätswerten der Plattform ist. In den meisten Fällen ist der CRTM im BIOS platziert (vgl. Abbildung 2.1) und wird vor dem eigentlichen BIOS ausgeführt, um so frühzeitig wie möglich das TPM mit Integritätswerten versorgen zu können.(vgl. [Mülo8, S.28])

Der RTR dient als Kernkomponente für die sichere Identifikation der Trusted Computing Platform (TCP) und die Absicherung der Integritäts- und Konfigurationswerte, wel-

che im Rahmen des TCG-Konzeptes der Remote Attestation (in Abschnitt 2.4.6.4 (Seite 34) beschrieben) zur Überprüfung des Systemzustandes an einen Kommunikationspartner übertragen werden. Die Identifikation der TCP wird durch einen so genannten Endorsement Key (EK) gewährleistet, welcher im nicht flüchtigen Speicher des TPMs gespeichert wird. (vgl. [Mülo8, S.28]) Der EK wird mit weiteren Schlüsseltypen am Ende dieses Abschnittes beschrieben.

Der RTS bildet die Basis für die sichere Speicherung von Schlüsselmaterial außerhalb des TPM. Der RTS besteht aus einem so genannten Storage Root Key (SRK), welcher im TPM gespeichert und durch den öffentlichen Teil des EKs des TPMs verschlüsselt und dadurch gesichert wird.(vgl. [Mülo8, S.29,]) Der SRK ist dabei der oberste (mit Ausnahme des EK) Schlüssel in einer prinzipiell beliebig langen Schlüsselhierarchie in Form eines Baumes für die gilt, dass ein Schlüssel einer Ebene $n + 1$ mit einem Schlüssel der Ebene n verschlüsselt wird. Die Vertrauenswürdigkeit eines Schlüssels einer beliebigen Ebene hängt damit von der Vertrauenswürdigkeit der (Eltern-) Schlüssel in der darüber liegenden Ebene ab. Damit ergibt sich für das Vertrauen in das gesamte Konstrukt, dass einem speziellen Schlüssel (z.B. Blatt-Knoten) genau dann vertraut wird, wenn allen Knoten auf dem Pfad zum obersten Schlüssel (Wurzel-Knoten) und diesem vertraut wird. Abbildung 2.2 (Seite 19) veranschaulicht die genannten Zusammenhänge der Vertrauensanker.

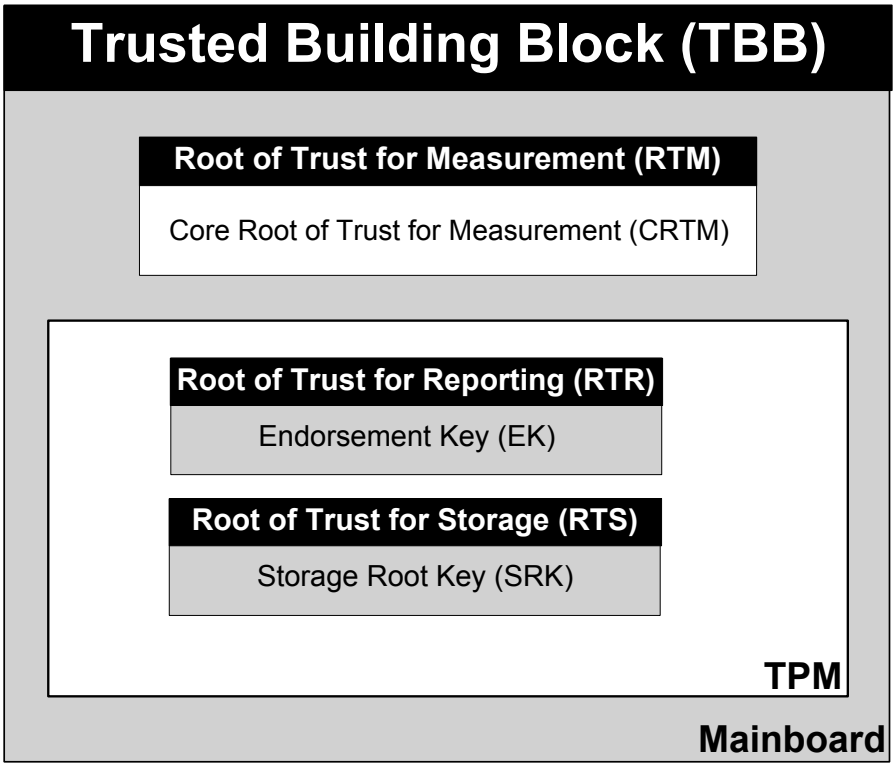

Abbildung 2.2 TBB in Anlehnung an Müller [Mülo8, Abb.4.2]

Schlüsseltypen

Die im Rahmen TCG-Konzepte genutzten verschiedenen Schlüsseltypen werden hier näher beschrieben. Dabei werden zunächst besondere Schlüsseltypen und danach die Eigenschaften der allgemeinen vom TPM bzw. TSS verwalteten Schlüsselkategorien erklärt. Die im Kontext verschiedener Schlüsseltypen zusätzlich benötigten Zertifikatstypen werden anschließend beschrieben. Übergreifend gilt, dass Schlüssel migrierbar, d.h. auf ein anderes TPM übertragbar oder nicht-migrierbar sein können. Zudem können die Verwendung und Migration (im Falle von migrierbaren Schlüsseln) von Schlüsseln an Passwörter gebunden werden. (vgl. [Mülo8, S.46]) Beispielsweise ist die Verwendung des EK durch das Owner Authorization Secret (OAS) beschränkt. (vgl. [Mülo8, S.37])

Endorsement Key (EK)

Beim EK handelt es sich um ein asymmetrisches 2048-Bit- RSA[1]-Schlüsselpaar (vgl. Abschnitt 2.3.2 (Seite 11)), welches im TPM generiert und im nicht flüchtigen Speicher dessen gespeichert wird. Der private Schlüsselteil verlässt dabei **niemals** das TPM. Beim EK handelt es sich folglich um ein nicht-migrierbares Schlüsselpaar. Dadurch, dass nur das konkrete TPM über den privaten Schlüsselanteil verfügen kann, stellt das EK-Schlüsselpaar die Identität dieses TPMs dar. Das Vertrauen in diese Identität des TPMs wird jedoch durch das Endorsement Credential (EKCRED) bzw. Platform Credential (PCRED) beeinflusst. Die TCG erwartet, dass pro TPM ein einziges EK-Schlüsselpaar benötigt wird. (vgl. [TCG07, S.11])

Storage Root Key (SRK)

Ein SRK ist analog zum EK ein asymmetrisches 2048-Bit- RSA[2]-Schlüsselpaar (vgl. Abschnitt 2.3.2 (Seite 11)). Der SRK kann im TPM erst generiert werden, nachdem ein EK vorhanden ist, da der SRK intern mit dem öffentlichen Teil des EKs verschlüsselt gespeichert wird. Zudem ist der SRK neben dem EK das einzige Schlüsselpaar, welches direkt im nicht flüchtigen Speicher des TPM gespeichert wird. (vgl. [Mülo8, S.34], [TCG07, S.18])

Attestation Identity Key (AIK)

Attestation Identity Keys (AIKs) sind nicht-migrierbare Signaturschlüsselpaare, die ausschließlich vom TPM zur Signatur von TPM-basierten Informationen genutzt werden. Sie können mit 512,1024 oder 2048 Bit im TPM generiert werden, die TCG empfiehlt jedoch 2048 Bit zu nutzen. Sie werden von der TCG auch als *Identity*-Schlüssel bezeichnet. (vgl. [Mülo8, S.46] , [TCG07, S.18]) Beispielsweise werden diese zur Signatur der Werte der Platform Configuration Registers (PCRs) (diese werden in Abschnitt 2.4.4 (Seite

[1] RSA82.
[2] RSA82.

24) näher erläutert) genutzt, um den Konfigurationsstatus einer TCP zu attestieren. (vgl. hierzu auch Abschnitt 2.4.6.4 (Seite 34))

Owner Authorization Secret (OAS)

Das OAS ist ein Kennwort, welches den Zugriff auf alle Befehle des TPM freischaltet. Es wird vom Besitzer des TPMs bzw. der TCP gewählt und durch das TPM in einen 160-Bit-SHA1 Hashwert überführt. Dieser wird mit dem öffentlichen Teil des EK verschlüsselt und im nicht-flüchtigen Speicher des TPM gespeichert. Es dient dem Besitzer der Plattform zur Authentifizierung am TPM bzw. zur Autorisierung der Ausführung von sicherheitskritischen Funktionen. (vgl. [Ric09, S.26],[Mül08, S.35,37])

Schlüsselkategorien

Neben den genannten spezielleren Schlüsseltypen gibt es noch eine Reihe von eher generischen Schlüsselklassen, von denen theoretisch beliebig viele (durch das Volumen der an die TCP angeschlossenen Datenträger begrenzt) Instanzen durch den TSS in Verbindung mit dem TPM gespeichert werden können. Dies geschieht gemäß der beim RTS bereits beschriebenen Schlüsselhierarchie. Die Kategorien werden im Folgenden kurz beschrieben und in der TCG-Spezifikation [TCG07, S.17,18] bzw. von Müller (vgl. [Mül08, S.46,47]) ausführlicher erklärt. Die TCG definiert beispielsweise eine Klasse von *legacy*-Schlüsseln, bei denen es sich um bereits außerhalb des TPM erzeugte symmetrische oder asymmetrische Schlüssel handelt, die aus diesem Grund auch zwingend migrierbar sind. Schlüssel dieser Kategorie können zudem sowohl zur Verschlüsselung als auch für Signaturen eingesetzt werden.

Weiterhin gibt es *Binding*-Schlüssel, bei denen es sich i. d. R. um asymmetrische Schlüsselpaare handelt, welche (sieh auch Abschnitt 2.4.6.2 (Seite 32)) zur direkten Verschlüsselung kleiner Datenmengen, wie zum Beispiel von symmetrischen Schlüsseln, genutzt werden können. *Binding*-Schlüssel können migrierbar oder nicht-migrierbar erzeugt werden.

Für Signaturzwecke gibt es zudem *Signing*-Schlüssel, welche ausschließlich für die Signatur beliebiger Daten genutzt werden können. Auch diese können migrierbar oder nicht-migrierbar angelegt werden.

Um auch größere Datenmengen außerhalb des TPM verschlüsseln zu können, gibt es *Storage*-Schlüssel. Diese asymmetrischen Schlüsselpaare dienen der direkten Verschlüsselung von kleinen und großen Datenmengen. Der bereits beschriebene SRK ist beispielsweise ein (spezieller) *Storage*-Schlüssel.

Um Sitzungsschlüssel für Kommunikationszwecke verwalten zu können, sieht die TCG außerdem noch die so genannten *Authentication*-Schlüssel vor. Bei diesen handelt es sich um symmetrische Schlüssel um die Ver- und Entschlüsselung von Kommunikationssitzungen zu gewährleisten. (vgl. [TCG07, S.18])

Zertifikatstypen

Die Eigenschaften der TCP bzw. der zugehörigen und bereits beschriebenen Schlüssel bzw. Schlüsseltypen werden mit verschiedenen Zertifikaten dokumentiert bzw. beglaubigt. Die verschiedenen Zertifikatstypen werden hier kurz beschrieben.

Endorsement Credential (EKCRED)

Das EKCRED ist vom Hersteller bzw. vom Ersteller des EK ausgestellt und soll bestätigen, dass der EK korrekt erstellt und in einem korrekten TPM nicht-migrierbar gespeichert ist. (vgl. [TCG07, S.10], [Mülo8, S.62]) Im EKCRED enthalten ist der Name des Herstellers, die Herstellerteilenummer und die Version bzw. Revision des TPMs sowie der öffentliche Schlüsselteil des EK. Da der öffentliche Schlüsselteil des EK-Schlüsselpaares im EKCRED enthalten ist, ist dieses ein, hinsichtlich des Schutzzieles Datenschutz, zu schützendes Objekt. Dies liegt in der Beschaffenheit des EKs begründet bzw. der Tatsache dass dieser (siehe oben) als eindeutige Identifikation eines TPMs dient. Das genannte Zertifikat gibt es dabei bedingt durch die maximale Anzahl von EKs pro TPM nur ein einziges Mal.

Conformance Credential (CCRED)

Das Conformance Credential (CCRED) beschreibt die Übereinstimmung des TBB-Designs der TCP mit den korrespondierenden TCG-Spezifikationen. Es wird von einem prüfenden Unternehmen bzw. einer prüfenden Organisation ausgestellt, welche damit bestätigt, dass es sich um ein TPM gemäß den TCG-Spezifikationen handelt und um eine TCP, welche gemäß eines korrekten TBB-Designs entworfen und implementiert wurde. Es kann ggf. für einzelne Komponenten des TBB einzelne CCREDs geben. Ein CCRED kann den Namen der prüfenden Organisation, den Namen des Herstellers der Plattform, die Modellbezeichnung und -version der Plattform sowie den Namen des Herstellers des verbauten TPMs und die Modellnummer bzw. -version von diesem enthalten. In den CCREDs sind keine Datenschutz-relevanten Daten enthalten, da diese keine Informationen über den EK und damit die Identität der TCP bzw. des TPMs beinhalten. (vgl. [TCG07, S.11])

Platform Credential (PCRED)

Das PCRED dient erweiternd zum EKCRED dazu, die gesamte TCP, also das das TPM umgebende System zu beschreiben. Dazu enthält es neben dem EKCRED auch alle nötigen CCREDs (bspw. für das TPM und die TCP) sowie bzgl. der Plattform der Name des Herstellers, die Modellnummer und die Version. Das PCRED enthält durch das enthaltene EKCRED Datenschutz-relevante Informationen und muss somit entsprechend geschützt werden. (vgl. [TCG07, S.11])

Validation Credential (VCRED)

Zusätzlich zu den PCREDs sollen Validation Credentials (VCREDs) die korrekte Funktionsweise von einzelnen Hardware- oder Softwarekomponenten beglaubigen. Die TCG erwartet dabei, dass die Hersteller dieser Komponenten entsprechende Referenzmessungen in Form von erwarteten Hashwerten z.B. in Reinräumen anfertigen und diese signiert mit ihren Produkten ausliefern. Dabei wird davon ausgegangen, dass eine Komponente

genau dann korrekt gemäß den Herstellerangaben funktioniert, wenn die Ermittlung einer Prüfsumme über diese der korrespondierenden Prüfsumme im VCRED entspricht. Allerdings beschränkt sich die Annahme der TCG auf wichtige Komponenten, die ein potentielles Risiko für die Sicherheit der TCP sein könnten. (vgl. [TCG07, S.12])

Attestation Identity Credential (AIC)

Die Attestation Identity Credentials (AICs) dienen der Beschreibung und Beglaubigung der Eigenschaften der TCP durch eine dritte vertrauenswürdige Instanz, welche ein AIC für einen AIK dieser TCP ausstellt. Im Gegensatz zum PCRED enthält ein AIC jedoch keine Datenschutz-relevanten Informationen, da der öffentliche Teil des EK nicht enthalten und somit keine Identifikation der TCP möglich ist. Bevor ein AIC jedoch ausgestellt wird, werden von der ausstellenden Instanz die PCREDs und VCREDs auf ihre Glaubwürdigkeit untersucht und getestet, dass es sich um das in den PCRED beschriebene TPM handelt. Dieser Vorgang wird in Abschnitt 2.4.6.4 (Seite 34) detailliert beschrieben. Ein AIC enthält im Wesentlichen den öffentlichen Teil des AIK, welcher durch das AIK beglaubigt werden soll, kann jedoch ggf. weitere Informationen enthalten. (vgl. [TCG07, S.13])

2.4.4 Trusted Platform Module

Aus den in Abschnitt 2.3.4 (Seite 13) und Abschnitt 2.4.1 (Seite 14) geschilderten Problemen heraus entwickelte die TCG eine Spezifikation für einen Hardwarechip, der als vertrauenswürdiger Hardware-Anker fungieren soll. Gemäß dieser Spezifikation stellen einige Chip-Hersteller, wie beispielsweise Infineon [Teco5], die so genannten TPMs her, die auf Mainboards verschiedener Hersteller integriert wurden. Der Chip-Hersteller Intel integrierte diese in eine interne Sicherheits- und Remotemanagementlösung, welche auf Funktionen zurückgreift, die in die Komponenten integriert wurden (vgl. [Cor11], [Gra07], [Mülo8, Abschnitt 5.1]).

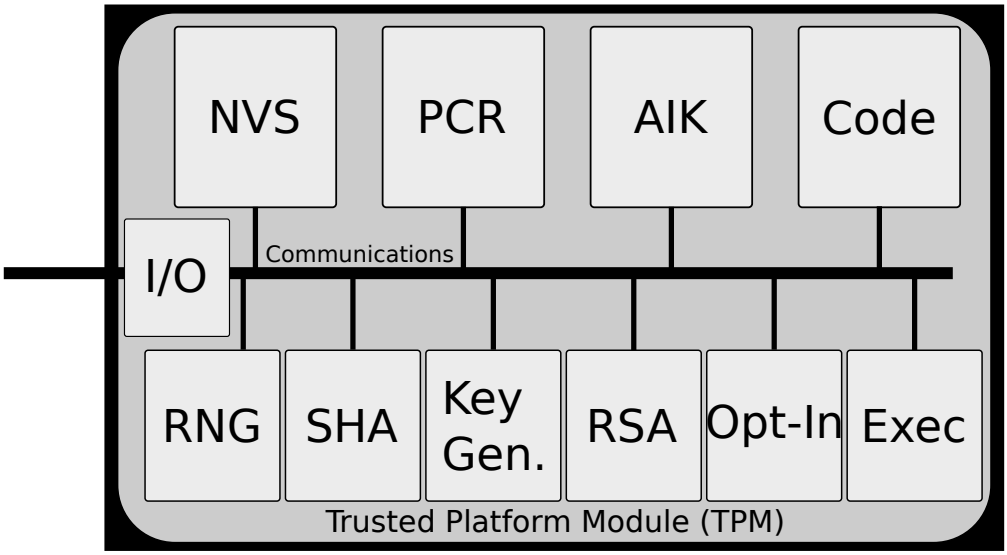

Abbildung 2.3 Architektur eines TPMs [TCG07, Abb. 4:g]

Abbildung 2.3 beschreibt die von der TCG spezifizierte Architektur eines TPMs. Im Folgenden wird daher erklärt, um was es sich bei einigen für diese Arbeit wichtigen Komponenten des TPMs handelt. Die nicht erläuterten Komponenten können in der Spezifikation nachgeschlagen werden. [TCG07, S.19,20] Bei der in Abbildung 2.3 mit *NVS* beschrifteten Komponente handelt es sich um einen nichtflüchtigen Speicher, der wie in Abschnitt 2.4.3 (Seite 16) beschrieben wichtige Dinge wie beispielsweise den EK und SRK im TPM speichert. Ein Zugriff von außen auf den nichtflüchtigen Speicher ist zudem nur durch die Funktionen des TPMs möglich. Die *RSA*- und *Key Gen.*-Komponente dient der Generierung und Nutzung von RSA-Schlüsseln mit bis zu 2048 Bit im TPM. (vgl.[TCG07, S.20]) Außerdem ist spezifiziert, dass ein TPM-Chip einen in Hardware realisierten Zufallszahlengenerator (vgl. Abbildung 2.3 *Random Number Generator (RNG)*) und ab TPM-Spezifikation 1.2 einen fälschungssicheren Zeitzähler (*Tick Counter*) hat, welcher nach Verlust bzw. neuem Anlegen der Stromversorgung automatisch mit einem Hersteller abhängigen initialen Wert (meist Null) beginnt. Um Hashwerte TPM-intern erzeugen zu können, besitzt ein TPM gemäß der TCG auch einen *SHA-1* Generator. (vgl. Abbildung 2.3) Zudem enthält ein TPM so genannte PCRs. Diese dienen der sicheren Erfassung von Systemkonfigurationsmesswerten und arbeiten nach dem einfachen Prinzip der Hash-Konkatination. Dabei kann ein PCR-Wert nicht überschrieben, son-

dern nur um einen weiteren Hash erweitert werden. Praktisch heißt dies, dass bei einer angeforderten Erweiterung eines PCRs der ursprüngliche Wert mit dem Wert um den er erweitert werden soll konkatiniert und das Ergebnis gehasht und im PCR hinterlegt wird. Die TCG drückt das mit folgender Schreibweise aus:
$PCR[n] \leftarrow SHA\text{-}1\ (PCR[n] + measured\ data)$ [TCG07, S.8].
Außerdem schreibt die TCG nicht vor, ob PCRs beim Start der TP zurückgesetzt werden müssen oder nicht. (vgl. [TCG07, S.19]) Aufgrund der begrenzten Speicherkapazitäten der TPMs spezifiziert die TCG die Nutzung eines Key Cache Managers (KCMs), um eine geringe Anzahl von temporären Schlüsselspeicherplätzen direkt im TPM vorzuhalten. In diesen sollen dann nur gerade bzw. häufig benötigte Schlüssel gespeichert bleiben, während alle anderen gemäß der Schlüsselhierarchie (vgl. Abschnitt 2.4.3 (Seite 16)) mit einem ihrer in der Hierarchie höher liegenden *Storage Keys* verschlüsselt auf externe Speichermedien ausgelagert werden. (vgl. [TCG07, Abb. 4:f,S.17,18])

Wie bereits in Abschnitt 2.4.2 (Seite 15) angedeutet, sind TPMs nicht selbst vor physischen Angriffen gesichert bzw. nur insofern, als dass physische Angriffe hinterher feststellbar sind. (vgl. [TCG07, S.21]) Dies hat den Grund, dass die Chips in der Herstellung und Verwendung möglichst günstig sein sollten, was ein weiteres Ziel der Spezifikation der TCG war, um eine hohe Verbreitung zu erreichen. (vgl. [Cha+07, S.13]) Weiterhin bietet ein TPM die Möglichkeit, durch einen hierarchischen Einsatz von Schlüsseln eine unbegrenzte (bzw. nur durch das Speichervolumen der am System angeschlossenen Speichermedien begrenzte) Anzahl von Schlüsseln verschlüsselt zu speichern und zu nutzen. Die durch das TPM erzeugten Schlüssel können dabei entweder migrierbar sein, um diese auf einen anderen PC übertragen zu können, oder nicht-migrierbar, um eindeutig einen konkreten PC bzw. dessen TPM identifizieren zu können. Für das Konzept nehmen wir hierbei an, wie in Abschnitt 1.2 (Seite 4) beschrieben, dass TPMs sicher sind. TPM-Chips sind in der Praxis derzeit am ehesten in PCs für geschäftliche Anwender zu finden.

Dennoch unterliegen die TPMs auch verschiedenen Beschränkungen wovon einige im Folgenden aufgelistet werden.

Beschränkungen:

Bei der Spezifizierung der TPM gab es Funktionen, die aus verschiedenen Gründen

nicht spezifiziert worden sind. So ist z. B. keine Echtzeit-Uhr im TPM-Chip enthalten, es gibt für diesen keine Batterie, um Informationen in flüchtigen Speichern zu erhalten, und auch auf eine Unterstützung von symmetrischer Verschlüsselung im TPM wurde verzichtet. Dies wurde darin begründet, dass einerseits das Konfliktpotential mit Exportrichtlinien verringert werden sollte und andererseits moderne symmetrische Algorithmen für eine schnelle Ausführung in Software ausgelegt sind. [Cha+07, S.25]
Des Weiteren ist das TPM auf eine sequentielle Ausführung von Kommandos und damit auf ein Kommando pro Zeiteinheit beschränkt (vgl.[Cha+07, S.63 „Talking to the TPM“]). Die Software-seitige Kommunikation mit dem TPM ist außerdem auf eine serielle Kommunikation mittels eines Gerätetreibers und lokaler Software beschränkt. [Cha+07, S.141]

2.4.5 TCG Software Stack

Wie in Abschnitt 2.4.2 (Seite 15) beschrieben, benötigen Anwendungsprogramme eine unterliegende Softwareschicht, die die Fähigkeiten des TPMs über ein Application Programming Interface (API) bereitstellen kann. Der TCG-Software Stack besteht selbst aus mehreren Schichten. Dabei unterteilen sich diese in Schichten, die im *kernel mode* bzw. *user mode* ausgeführt werden.

Der TPM-Gerätetreiber bildet die Basis und stellt den einzigen Bestandteil des TSS dar, welcher im *kernel mode* läuft. (vgl. [TCG07, Abb. 4:i]) In der TCG 1.1b Spezifikation [TCG02] ist seitens der TCG noch keine Geräteschnittstelle definiert worden. Dies wurde in der darauf folgenden Version 1.2 [TCG05] nachgeholt, da ansonsten die Geräteschnittstellen stark Hersteller abhängig waren (vgl. [Cha+07, S.45] bzw. [TCG11c]).

2.4.5.1 TCG Device Driver Library (TDDL)

Die Hauptaufgabe der TCG Device Driver Library (TDDL) besteht laut [TCG07, S.26] darin, Anwendungsprogrammen eine Betriebssystem- und Hersteller-übergreifende einheitliche Schnittstelle im *user mode* bereit zu stellen und Herstellern die Möglichkeit zu geben, für diese Schnittstelle einen Simulator des eigenen TPM-Chips zu entwickeln. Eine freie Implementierung eines Simulators gemäß der TCG 1.2 Spezifikation ([TCG11d], [TCG11e], [TCG11f]) wird in [Str04] beschrieben und steht online zur Verfügung.(siehe [SSM10])

Dabei ist der Zugriff auf das TPM nur mit einem Prozess zur Zeit möglich, wie in [TCG07, S.26] beschrieben wird: "It does not manage threaded interactions with the TPM. [...] Since the TPM is not multithreaded, there would be a single-instance of TDDL, per platform, and it enforces single threaded access to the TPM."

2.4.5.2 TCG Core Services (TCS)

Die TCG Core Services (TCS) stellen TCG Service Providern (TSP) fünf grundlegende Dienste bereit, um sicherzustellen, dass grundlegende Funktionalitäten des TPMs mit dem spezifizierten Verhalten genutzt werden können. Dazu gehört, dass mit dieser Schicht, durch ein Kontext-Management (vgl. [TCG07, S.26], [Cha+07, S.78,144]) erstmalig (im Gegensatz zu Abschnitt 2.4.5.1) ein nebenläufiger Zugriff auf das TPM ermöglicht wird. Damit werden einige der Beschränkungen in Abschnitt 2.4.4 (Seite 26) addressiert. Die TCS synchronisieren und verwalteten die Zugriffe bzw. Sitzungen der verschiedenen TSP und die sich daraus ergebenden Implikationen beim TPM. Beispielsweise müssen, aufgrund der beschränkten internen Speicherkapazität des TPMs Schlüssel je nach TSP in das TPM geladen bzw. ausgetauscht werden (vgl. [Cha+07, S.78] und [TCG07, S.16 Abschnitt 4.2.7]). Auch wird mittels der TCS ermöglicht, hinter diese einen Remote Procedure Call (RPC)-Server zu schalten und damit entfernte TSP anzubinden und damit die Beschränkung auf lokale Anwendungen aufzuheben. [TCG07, S.27 Abb. 4:j]

2.4.5.3 TCG Service Provider (TSP)

Die TSP setzen auf den TCS auf und stellen Anwendungen C-Schnittstellen zur Verfügung, welche direkt im Programm-Quelltext genutzt werden können. Ein TSP stellt dabei die gesamten Funktionalitäten des TPMs bereit bzw. ergänzt diese ggf. um eigene Erweiterungen. Als Beispiel hierfür werden von Challenger et. al z. B. Schlüsselspeicher und Dialoge für Autorisationsdaten genannt. [Cha+07, S.79]

Wie in [TCG07, S.27 Abb. 4:j] zu sehen ist bieten TSP in Verbindung mit den TCS außerdem die Möglichkeit bisherigen Kryptographiestrukturen bzw. -anwendungen die Fähigkeiten des TPMs bereitzustellen bzw. diese zu integrieren.

2.4.6 Trusted Computing-Konzepte

Im Folgenden Abschnitt werden einige, im zu Grunde liegenden Konzept von Richter verwendete Konzepte der TCG, welche im Kontext des *Trusted Computings* genutzt werden, in Kurzform erläutert. Zunächst wird ein sicherer bzw. vertrauenswürdiger Startvorgang, darauf aufbauend eine Beschreibung eines sicheren Nachrichtenaustausches (in Abschnitt 2.4.6.2 (Seite 32)), eines Konzeptes um Nachrichten bzw. Daten an *Zeitsitzungen* eines TPMs zu binden (in Abschnitt 2.4.6.3 (Seite 33)) und eines Konzeptes für die sichere bzw. vertrauenswürdige Zusicherung von Eigenschaften einer TCP (in Abschnitt 2.4.6.4 (Seite 34)).

2.4.6.1 Trusted Boot

Das TCG-Konzept *Trusted Boot* bildet die Basis einiger weiterer TCG-Konzepte, da diese i. d. R. einen vertrauenswürdigen Zustand der TCP voraussetzen. Die Grundidee besteht darin, wie in Abschnitt 2.4.3 (Seite 16) beschrieben, von vertrauenswürdigen Komponenten bzw. Ausführungszuständen schrittweise durch das Erzeugen und sichere Dokumentieren von Integritätsmessungen einen vertrauenswürdigen Systemstart zu ermögli-

chen. Die sichere Dokumentation der Integritätsmessungen übernimmt dabei das TPM bzw. wird durch die PCR des TPMs sichergestellt. Nach Müller und Smith [Mülo8, S.74] unterscheidet sich ein *Trusted Boot* von einem *Secure Boot* insofern, als dass bei einem *Secure Boot* nicht nur die Messwerte für eine Beurteilung der Integrität ermittelt und fälschungssicher gespeichert werden, sondern der Startvorgang bei nicht übereinstimmenden Integritätswerten gestoppt wird. Ein beispielhafter Ablauf eines solchen vertrauenswürdigen *Trusted Boot* Startprozesses wird von Reid in [RCo5, Abschnitt 4.2] geschildert. Dabei wird zunächst, in Abbildung 2.4 (Seite 31) im ersten Schritt zu sehen, der CRTM im BIOS ausgeführt und erzeugt eine Prüfsumme über den ausführbaren Teil des BIOS. Die Prüfsumme wird im zweiten Schritt in einem PCR des TPMs gespeichert und der Ausführungspfad dann an das BIOS übergeben. In den folgenden Schritten fünf bis neun werden Prüfsummen für die *Option ROMs* von Steckkarten, *CPU Microcode Updates* und schließlich für den zu ladenden *OS Loader* erzeugt und jeweils in PCRs oder ggf. in ein einiges PCR des TPMs hinzugefügt. Im zehnten Schritt wird dann die Kontrolle an den *OS Loader* übergeben.

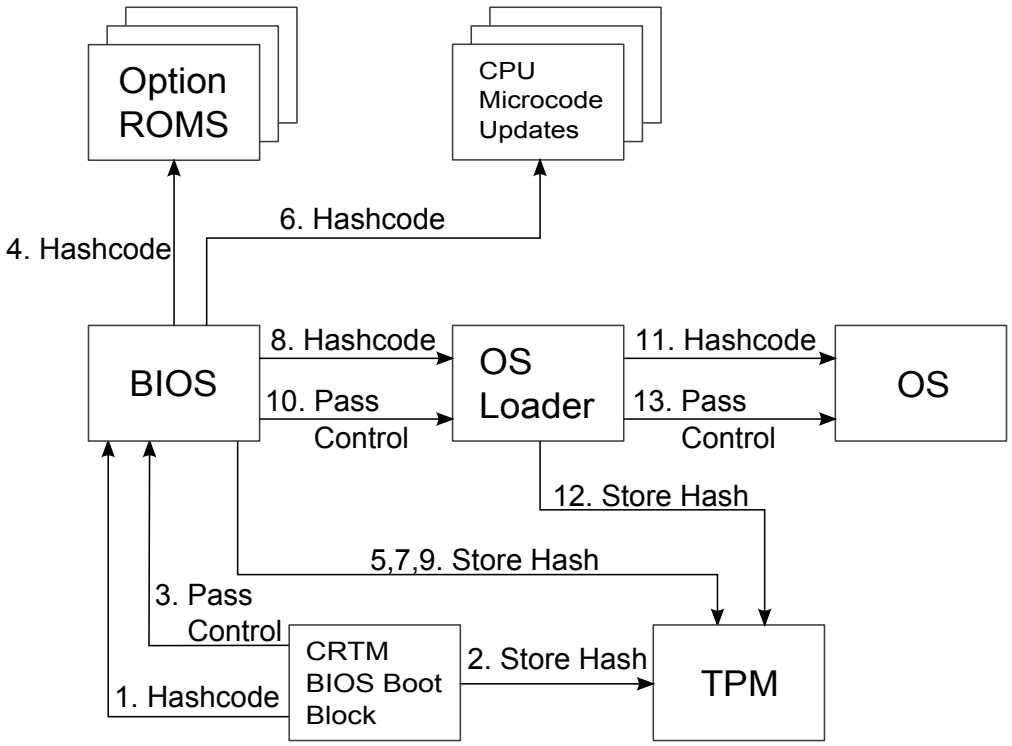

Abbildung 2.4 Beispielhafter Ablauf eines *Trusted Boot*-Prozesses

Dieser Teil der Vertrauenskette wird von Müller als *Static Chain of Trust* bezeichnet, wie in Abbildung 2.5 (Seite 32) zu sehen ist. Dabei bestehen nach Müller nur für diesen Teil der gesamten Vertrauenskette Spezifikationsaussagen seitens der TCG. Es fehlen daher konkrete Vorgaben, welche Teile eines Betriebssystems in die Messung eines vertrauenswürdigen Systems einbezogen werden müssen. In der zu Grunde liegenden Arbeit von Richter wird hier auf verschiedene Projekte, wie z.B. Trusted Grub [Sir11], IBM IMA [IBM11] oder BIND [SPD05], verwiesen, welche Lösungsansätze für die *Dynamic Chain of Trust* nach Müller bieten sollen. Eine genauere Betrachtung dieser ist auch in [Mülo8, Kapitel 8] zu finden.

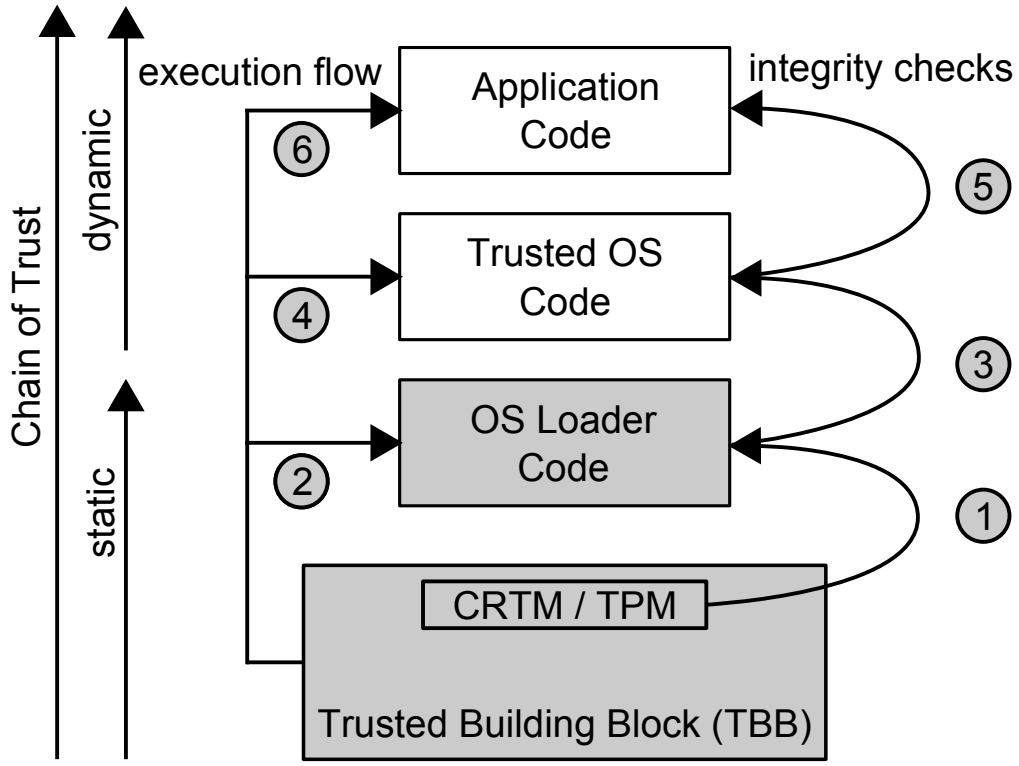

Abbildung 2.5 Chain of Trust nach Müller [Mülo8, Abb 4.14]

2.4.6.2 Trusted Messaging

Um eine sichere Kommunikation zu ermöglichen, hat die TCG vier verschiedene Klassen von Nachrichten "Binding", "Signing", "Sealing" und "Sealed Signing" für einen sicheren Nachrichtenaustausch spezifiziert. [TCGo7, S.15] Diese werden nachfolgend erläutert :

"Binding"

dient dabei der asymmetrischen Verschlüsselung von Nachrichten, mit dem Zweck diese nur mit einem privaten Schlüssel, welcher vom TPM bzw. den TCS oder TSP verwaltet wird und an ein TPM gebunden ist, entschlüsseln zu können. Wenn dieser Schlüssel beim TPM als nicht-migrierbar angelegt wurde, kann die Entschlüsselung der Nachricht an das TPM und damit an die vertrauenswürdige Plattform gebunden werden (vgl.[TCGo7, S.15]).

“Signing”

bezieht sich dabei im Wesentlichen darauf, dass es im TPM die Möglichkeit gibt, Schlüssel anzulegen, deren Verwendung lediglich zu Signaturzwecken (vgl. Abschnitt 2.3.3 (Seite 12)) und nicht für normale Verschlüsselung erlaubt ist.

“Sealing”

ist eine Erweiterung des “Binding´s” und bietet eine Bindung der Nachricht an eine spezielle Plattform mit gewünschter Konfiguration. Dabei wird eine Nachricht symmetrisch verschlüsselt, der dafür genutzte Schlüssel wird zusammen mit einer geforderten Plattform-Konfiguration in Form von PCR-Werten asymmetrisch für einen nicht-migrierbaren privaten Schlüssel des Empfängers verschlüsselt und kann bei der empfangenden Plattform nur dann entschlüsselt werden, wenn die vorgegebene Systemkonfiguration besteht. D.h. dass die Nachricht nur dann entschlüsselt werden kann (vgl.[TCG07, S.15,16]).

“Sealed Signing”

ist eine Erweiterung des “Signing”, um zusätzlich zu einer Signatur gesicherte Informationen über den Status der Systemkonfiguration zum Signaturzeitpunkt zu erhalten. Bei der Signatur werden die von einem Kommunikationspartner zur Verifizierung gewünschten PCR-Werte mit in die Nachricht bzw. Berechnung des Hashwertes einbezogen(vgl.[TCG07, S.16]).

2.4.6.3 Trusted TickStamping

Die von der TCG spezifizierten TPMs (vgl. Abschnitt 2.4.4 (Seite 24)) müssen nach Spezifikation nicht zwingend über eine interne Batterie dauerhaft mit Strom versorgt werden. [TCG11d, S.108] Da deshalb eine sichere Uhr innerhalb des TPMs als zu teuer befunden wurde, ist im Rahmen der Version 1.2 der TCG-TPM-Spezifikation ein so genannter *Tick Counter* eingeführt worden. Dieser stellt zwar keine reale Uhrzeit zur Verfügung, wird jedoch bei jedem Start des TPMs zurückgesetzt auf einen Startwert von null. Zudem wird TPM-intern vom RNG eine Nonce generiert und mit dem *Tick Counter* in einer *Timing Session* verknüpft. Die so erzeugte *Timing Session* bietet die Möglichkeit, Daten gemeinsam mit der Nonce der *Timing Session* und dem aktuellen Wert des

Tick Counters zu signieren. Da der *Tick Counter* und die Nonce nicht von außerhalb des TPMs beeinflusst werden können, verfügen TPMs ab Version 1.2 über eine interne Uhr, die lediglich noch mit einer realen Uhrzeit (von der TCG werden dafür Uhrzeiten in UTC empfohlen) synchronisiert werden muss. (vgl. [Cha+07, S.276],[TCG11d, S.104,105])

2.4.6.4 Trusted Platform Attestation

Wie in Abschnitt 2.4.3 (Seite 16) erwähnt, ist eine der wichtigsten Funktionalitäten die sichere und glaubwürdige Zusicherung von Eigenschaften des TPMs bzw. der mit dem TPM ausgestatteten Plattform. Es wird seitens der TCG unterschieden zwischen der Zusicherung darüber, welche Informationen ein TPM hat "Attestation by the TPM", dass eine Plattform in der Lage ist, seine Konfiguration vertrauenswürdig zuzusichern "Attestation to the platform", die Konfiguration einer Plattform "Attestation of the platform" und die Authentifizierung einer Identität "Authentication of the platform" (vgl. [TCG07, S.5,6]). Diese basieren grundlegend auf der Signatur von Daten mit AIKs und der Übermittlung von AICs für diese an den Kommunikationspartner, welcher die Eigenschaften überprüfen möchte. Für die Anforderung eines korrespondierenden AIC zu einem AIK ist dabei nach dessen Erstellung eine Kommunikation mit einer dritten vertrauenswürdigen Partei einer so genannten Privacy Certification Authority (PCA) nötig, um über den öffentlichen Teil des AIKs ein Zertifikat der PCA zu erhalten, welches beispielsweise nach einer erfolgreichen Prüfung als Teil des AICs beglaubigt, dass das ausstellende TPM des AIKs ein Spezifikations-konformes TPM ist und der AIK an einen gültigen EK bzw. ein PCRED und CCRED gebunden ist (vgl. [Eck09, S.606,607], [TCG07, S.13]). Dabei ist jedoch in dem von der PCA ausgestellten AIC nicht der öffentliche Teil des EK enthalten, da dieser wie in Abschnitt 2.4.3 (Seite 16) beschrieben die Identität des TPM und damit des Systems ist und diese einem Kommunikationspartner, welcher nur wissen müsste, ob ein Teil der Systemkonfiguration die Anforderungen erfüllt, offenlegen würde. Aus diesem Grunde wird das Prinzip der Attestierung via AIK in [Eck09, S.601] auch als Pseudomisierungs-Konzept bezeichnet. Die bei den beschriebenen unterschiedlichen Zusicherungsformen verwendeten bzw. signierten Daten werden im Folgenden kurz aufgelistet.

“Attestation by the tpm”:

Hier werden die Daten, deren Existenz oder Kenntnis das TPM zusichern soll, mittels des AIK signiert und die Signatur zusammen mit dem AIC dem Kommunikationspartner übermittelt, welcher die Zusicherung verlangt hat. Dieser kann dann auf Basis des im AIC enthaltenen und von der PCA ausgestellten Zertifikates des öffentlichen Schlüssels des AIK, so die PCA auf seiner Liste der vertrauenswürdigen Zertifizierungstellen steht, überprüfen ob es sich um ein TPM handelt und der zugehörige Signaturschlüssel (AIK) an dieses gebunden ist und dann ob die Daten gemäß seinen Richtlinien korrekt sind (vgl. [TCG07, S.6]).

“Attestation to the platform”:

Hier wird ein AIK und ein zugehöriges AIC erzeugt bzw. angefordert, um zusichern zu können, das die TP in der Lage ist ihre Konfiguration vertrauenswürdig auszuweisen. Das AIC beglaubigt in diesem Zusammenhang die vom Hersteller zugesicherten Fähigkeiten des TPMs, welche in Form eines PCREDs bzw. CCREDs vorliegen.Dazu werden der PCA neben dem öffentlichen EK zur Identifizierung auch das vom Hersteller ausgestellte PCRED bzw. CCRED übermittelt (vgl. [TCG07, S.6]).

“Attestation of the platform”:

Um die Konfiguration eines Systems zuzusichern, werden die Werte der PCRs mit einem AIK signiert und gemeinsam mit dem zugehörigen AIC an den Kommunikationspartner zur Verifikation geschickt (vgl. [TCG07, S.6]).

“Authentication of the platform”:

Bei der Authentifizierung wird ein nicht-migrierbarer Signaturschlüssel mit einem AIK signiert und diese Signatur mit dem zugehörigen AIC übermittelt (vgl. [TCG07, S.6]).

Das ursprünglich einzige Verfahren für die Attestierung (in Version 1.1 der Spezifikation [TCG02]), welches, wie oben beschrieben, die Kommunikation mit einer vertrauenswürdigen dritten Partei der PCA vorsieht, stellte sich im Laufe der Zeit als problematisch heraus. So wird in [Cam04b, Folie 11] beschrieben, dass die PCA im Konzept der Attestierung einen Flaschenhals darstellt, da alle neuen AIKs von ihr nach Prüfung des TPM signiert werden müssen und eine nicht durchgehende Verfügbarkeit dieser somit zu einem Problem bei der Nutzung der Trusted Platform werden würde. Auch muss

die PCA einen hohen Schutz aufweisen, da eine kompromittierte PCA eine Zuordnung der, vermeintlich Anonymität versprechenden, AIKs zu den EKs ermöglichen würde. In [Eck09, S.608] und [Cam04b, Folie 11] wird außerdem das Problem beschrieben, welches entsteht, wenn der Kommunikationspartner, welcher eine Verifikation der Systemkonfiguration verlangt, und die PCA zusammenarbeiten, da dadurch die Wirkung der AIKs aufgehoben wäre und die Aufzeichnung eines Nutzungsprofils des mit dem TPM ausgestatteten Systems ermöglicht würde. In [Cam04b, Folie 11] wird weiter beschrieben, dass es für eine PCA kein wirkliches Geschäftsmodell gibt. Dies dürfte in der Praxis wirklich vertrauenswürdige PCAs verhindern, da aufgrund der hohen Schutzanforderungen ein Betrieb eines solchen Services entsprechend kostspielig sein dürfte. Wie oben beschrieben, würde eine Zusammenarbeit mit den Service-Anbietern, welche eine Zusicherung verlangen, den Datenschutz unterwandern, und da die Anbieter der PCA vertrauen müssen, können gemeinnützige Institutionen ebenfalls keine PCA betreiben, die von den Anbietern akzeptiert wird (vgl. [Cam04b, Folie 11], [BCC04, S.2] und [Cam04a, S.2]).

Aus den genannten Problemen heraus entwickelten Bricknel, Camenisch und Chen im Jahre 2004 das Konzept der "Direct Anonymous Attestation (DAA)"[BCC04], welches später Einzug in die TCG-Spezifikation 1.2 [TCG11d, S.150] fand. In [Ric09, S.30] wird dies so zusammengefasst, dass der Kommunikationspartner, welcher eine Zusicherung verlangt, nur wissen muss, dass es sich um ein TPM handelt, nicht jedoch um welches. In [Cam04b, S.13] wird die Grundidee der DAA so beschrieben, dass nicht ein Zertifikat ausgetauscht werden soll, sondern lediglich ein kryptografischer Beweis "cryptographic proof" erbracht werden soll, dass das TPM ein DAA-Zertifikat besitzt. Weiter soll so auch das Flaschenhals-Problem gelöst werden, da ein DAA-Zertifikat nur einmal ausgestellt und später nur bei einer Attestierungsanfrage bewiesen werden muss, dass das TPM dieses besitzt und eine Signatur über den AIK, den anfragenden Kommunikationspartner und die Zeit erstellt hat. Dies basiert auf dem Camenisch-Lysyanskaya Signaturschema, welches in [Cam04a, S.6] bzw. [Cam04b, ab Folie 15] beschrieben wird.

2.5 Sicherheitsprotokolle

Anders als bei Netzprotokollen handelt es sich bei Sicherheitsprotokollen weniger um eine Bit-genaue Beschreibung der Darstellung von Nachrichten eines Kommunikationsablaufes, sondern um eine genaue Beschreibung der Inhalte eines Kommunikationsvorganges. Dabei ist der wichtigste Grundsatz, dass der Kommunikationskanal im Zweifel unter Kontrolle eines möglichen Angreifers und daher unsicher hinsichtlich der Schutzziele ist. Sicherheitsprotokolle haben dabei das Ziel, ein oder mehrere Schutzziele (wie in Abschnitt 2.1 (Seite 8) beschrieben) in einer Kommunikation zu erreichen. In den meisten Fällen soll dabei ein sicherer Schlüsselaustausch bzw. eine sichere Authentisierung erreicht werden.

In [Ando8, S.13] werden Sicherheitsprotokolle von Ross Anderson wie folgt beschrieben :

> A typical security system consists of a number of principals such as people, companies, computers, and magnetic card readers, which communicate using a variety of channels including phones, email, radio, infrared, and by carrying data on physical devices such as bank cards and transport tickets. The security protocols are the rules that govern these communications. They are typically designed so that the system will survive malicious acts such as people telling lies on the phone, hostile governments jamming radio, or forgers altering the data on train tickets. Protection against all possible attacks is often too expensive, so protocols are typically designed under certain assumptions about the threats.[Ando8, S.13]

Ein wichtiger Faktor bei Sicherheitsprotokollen sind die Annahmen, denen sie unterliegen. So schreibt Anderson z. B. “A clear understanding of trust assumptions and their consequences is at the heart of security protocol design.”[Ando8, S.21] bzw. “an important point: that the correctness of a security protocol depends on the assumptions made about the requirements.”[Ando8, S.22]. Dies ist auch der Grund weshalb Sicherheitsprotokolle nicht ohne Prüfung der Annahmen bzw. Anpassung des Protokolls gemäß den aktualisierten Annahmen für einen neuen Anwendungszweck bzw. eine verän-

derte Umgebung eingesetzt werden können, wie Anderson schreibt : “A very common cause of protocol failure is that the environment changes, so that assumptions that were originally true no longer hold, and the security protocols cannot cope with the new threats.”[And08, S.22]

Netzprotokolle wie z. B. Secure Sockets Layer (SSL) oder Transfer Layer Security (TLS) setzen jedoch Teile oder gar mehrere Sicherheitsprotokolle in der Realität um.

2.6 Formale Notation

Da die in Abschnitt 2.5 (Seite 37) beschriebenen Sicherheitsprotokolle verschiedene Schutzziele (vgl. Abschnitt 2.1 (Seite 8)) zwischen verschiedenen Kommunikationspartnern sicher stellen sollen, ist ein umfassendes Verständnis des Kommunikationsvorganges mit allen wichtigen Inhalten unumgänglich, um sicherzustellen, dass sich keine Fehler im Design eingeschlichen haben. Da sich jedoch textuelle Beschreibungen im allgemeinen über viele Seiten oder sogar Kapitel erstrecken können und manche Teilbereiche eines Sicherheitsprotokolle ggf. doppelt mit unterschiedlichem Detail beschrieben werden, ist eine Analyse dieser aufwendig, und in der Praxis werden daher oftmals Fehler übersehen.

Einige der bekannteren oder größeren Fehler haben Anderson und Needham in ihrer bekannten Ausarbeitung “Programming Satan’s Computer”[AN95b] beschrieben. Um die Fehler in den Sicherheitsprotokollen dabei klar und deutlich aufzeigen zu können, haben sie diese auf wenige Zeilen in eine Notation vereinfacht, die sie bereits in einer Ausarbeitung, in der sie Regeln zur Entwicklung von Sicherheitsprotokollen aufstellten, einführten [AN95a]. Da diese Notation auch in der Literatur inzwischen weit verbreitet ist, wird diese auch hier genutzt und im Folgenden kurz vorgestellt werden. Die Grundidee dabei ist, dass es eine Reihe von Nachrichten, beteiligten Kommunikationspartnern und Schritten gibt, bis ein bzw. mehrere Schutzziele erreicht werden.

Pro Schritt wird dabei nur eine Kommunikation zwischen zwei Partnern beschrieben und die beteiligten Partner werden mit Großbuchstaben und die Richtung des Nach-

richtenaustausches mit einem → visualisiert.Zusätzlich werden unverschlüsselte Nachrichten mit **M**, Zeitstempel mit **T** und Nonces mit **N** dargestellt. Dabei haben diese i. d. R. Indizes zur Kennzeichnung woher diese stammen und um somit die spätere Übersichtlichkeit zu erhöhen. Kryptografische Schlüssel werden durch **K** dargestellt und ebenfalls durch Indizes zweier oder mehrerer Kommunikationspartner als symmetrische Schlüssel, die diesen bekannt sind beschrieben oder aber mit einem einfachen Indize als öffentlicher Schlüsselteil des Partners, dessen Indizes verwendet wurde gekennzeichnet. Ein privater Schlüsselteil wird dabei mit einem K^{-1} und einem Indizes gekennzeichnet. Um die Übersichtlichkeit und Fähigkeit dieser Notation, die Abläufe effizient darzustellen, zu zeigen, werden im Folgenden ein paar kleinere Beispiele beschrieben.

Beispielsweise benötigt eine symmetrische Kommunikation zwischen den Kommunikationsteilnehmern **A** und **B**, so der gemeinsame Schlüssel K_{AB} vorher bereits sicher ausgetauscht wurde lediglich ein bzw. zwei Schritte.

$$A \rightarrow B \quad : \quad \{B, M\}_{K_{AB}} \tag{2.1}$$
$$B \rightarrow A \quad : \quad \{A, M\}_{K_{AB}} \tag{2.2}$$

Oder die in Abschnitt 2.3.2 (Seite 11) beschriebene asymmetrische Kryptographie beschränkt sich auf:

$$A \rightarrow B \quad : \quad \{N_A\}_{K_B} \tag{2.3}$$
$$B \rightarrow A \quad : \quad \{N_A, N_B\}_{K_A} \tag{2.4}$$
$$\tag{2.5}$$

Dabei kann **B** im ersten Schritt mittels seinem privaten Schlüssel ${K^{-1}}_B$ N_A entschlüsseln und diese Nonce zu **A** in Schritt 2 verschlüsselt mit dessen öffentlichen Schlüssel K_A schicken. **A** kann dann ebenso wie **B** die Nonce N_A entschlüsseln und überprüfen, ob diese identisch mit der Nonce ist, welche **A** im ersten Schritt an B verschickt hat.

2.7 Rechtsgrundlagen

Die zum Verständnis des in [Ric09] vorgestellten Konzeptes nötigen Gesetze werden hier im Folgenden exemplarisch für die Bundesrepublik Deutschland kurz genannt bzw. ihre Wirkung beschrieben. Dies ist dabei allerdings informell zu verstehen und bietet lediglich eine Interpretation der vorliegenden Gesetzestexte bzw. Kommentaren Fachkundiger zu diesen. In [Ric09, Abschnitt 4.4.1 S. 59] werden zudem noch weitere zu Grunde liegende Gesetze und Richtlinien der europäischen Union bzw. anderer Länder betrachtet.

Da im Konzept [Ric09] digitale Informationen automatisiert erhoben werden sollen, welche als elektronische Beweisdokumente den Anforderungen der Gerichtsbarkeit standhalten sollen, müssen diese gemäß [BMJ11c, §371 a] mit einer qualifizierten elektronischen Signatur gemäß [BMJ09c, §2] versehen sein, um hinsichtlich der Beweiskraft Urkunden gleich gestellt zu sein. Andernfalls besteht nur die Möglichkeit einer Einführung als Augenscheinsobjekte gemäß [BMJ11c, §371 Abs.1], welches jedoch in der Praxis häufig zu einer Ablehnung deren führt (vgl. [Hof06], [Roß06, S. 5]).

In [Tüb04, S.1,2] wird das Recht zur Verkehrsüberwachung gemäß [BMJ10b, §49] bzw. [BMJ09b, §53] beschrieben.

In die automatisierte Erfassung von Daten spielen jedoch auch datenschutzrechtliche Vorgaben. So wird in [BMJ09a, §1] der Schutz des Persönlichkeitsrechtes des Einzelnen auf Basis des Urteils zur Volkszählung des Bundesverfassungsgerichts von 1983 [BVe83] welches auf einer Interpretation des [BMJ10a, Artikel 2 Absatz 1] basiert (vgl. [Fer07]) geregelt.

Auf Basis der strengen Datenschutzrechte in Deutschland hat zudem das Bundesverfassungsgericht im Jahre 2009 klar gemacht, dass eine grundlose Aufzeichnung der Verkehrsteilnehmer in Deutschland im Gegensatz zu deren Persönlichkeitsrecht gemäß [BMJ10a, Artikel 2 Absatz 1] steht. Ferner bezieht sich dies auf jegliche Verkehrsüberwachungsmaßnahmen, welche die Verkehrsteilnehmer aufnehmen bevor bzw. ohne dass diese eine Gesetzeswidrigkeit begangen haben (vgl. [Kru09]).

In [Ric09, S.46] wird zudem darauf eingegangen, dass der Versuch der Manipulation bzw. Störung von Verkehrsüberwachungsgeräten bzw. das Mitführen eines Gerätes, welches

dies ermöglicht, gemäß [BMJ10b, §23 Absatz 1b] eine Ordnungswidrigkeit darstellt und mit einem Bußgeld bestraft werden kann.

Die für ein Geschwindigkeitsüberwachungsgerät verbindlichen Eigenschaften werden in [PTB06] spezifiziert.

Die Archivierungsfristen von Verstößen gegen die Straßenverkehrsordnung [BMJ10b], welche im zentralen Melderegister in Flensburg gespeichert werden, liegen zwischen zwei und zehn Jahren. Dabei werden nur Ordnungswidrigkeiten nach §24 zwei Jahre, weitere Fälle ohne Führerscheinentzug bis zu fünf Jahre und alle weiteren Fälle zehn Jahre gespeichert. (vgl. [BMJ11b, §29]) Eine Ausnahme hiervon sind Fälle, in denen ein dauerhaftes Führerscheinverbot ausgesprochen wird. Ansonsten korrespondiert die Speicherungsdauer von maximal zehn Jahren bei Erwachsenen auch mit den Bestimmungen der Strafprozessordnung. (vgl. [BMJ01, §489])

Kapitel 3

Konzept

Wie in Abschnitt 1.4 (Seite 6) (*Vorgehensweise*) erläutert, wird in diesem Kapitel das Konzept von Richter [Ric09] zunächst beschrieben bzw. erklärt. Dabei wird zuerst der Kontext dargelegt und anschließend die daraus resultierenden Anforderungen an das Konzept. Danach folgt eine komplette Beschreibung des Funktionsablaufes im Konzept [Ric09], diese wird dann mit an die ursprüngliche Arbeit [Ric09] angelehnten Grafiken vereinfacht. Diese Form der Darstellung, welche auch in [Ric09] gewählt wurde, wird ferner einer kritischen Betrachtung unterzogen um Unklarheiten im Ablauf aufzuzeigen.

3.1 Beschreibung

Im Folgenden Abschnitt soll das Konzept von Richter [Ric09] zusammenfassend vorgestellt werden. Das Konzept wird aufgrund der verschiedenen Unklarheiten und Widersprüche, von denen in Abschnitt 3.2 (Seite 59) einige exemplarisch aufgezeigt werden, nach besten Wissen beschrieben. Dabei wird stets versucht, der tatsächlichen Intention des ursprünglichen Konzeptes möglichst nahe zu kommen. Zudem werden die Beschreibungen der zusammenfassenden von Richter et. al [RKR10] und einer weiterführenden Publikation von Kuntze und Rudolf [KR11] verwendet, um der ursprünglichen Intention möglichst nahe zu kommen.

Das von Richter beschriebene Konzept soll die Nichtabstreitbarkeit (vgl. Abschnitt 2.1 (Seite 8)) von realen Ereignissen, welche mittels automatischer digitaler Messungen dokumentiert worden sind, sicher stellen. Ein Hauptziel der Arbeit von Richter ist es, dies durch den Einsatz von TCG-Komponenten und -Konzepten zu erreichen, wie Richter schreibt : “The main purpose of this work is to [...] introduce Trusted Computing as solution for securing digital evidence in an automated process.” [Ric09, S.65] Da eine generische Beschreibung wenig anschaulich und verständlich ist, wird das Konzept anhand eines Anwendungsbeispieles erläutert. Zu diesem Zweck wird in der zu Grunde liegenden Arbeit ein Traffic Management System (TMS) verwendet, da dies ein anschauliches Beispiel aus der realen Welt liefert, in dem sichere digitale Beweise notwendig sind. Das TMS soll dabei in Form eines kleinen autarken eingebetteten Systems realisiert werden,

welches ein integriertes TPM besitzt, um die Fähigkeiten des TPMs zur Absicherung der digitalen Repräsentation eines realen Ereignisses nutzen zu können. Dieses kann beispielsweise die Geschwindigkeit der Verkehrsteilnehmer, den Abstand zueinander, Rotlichtverstöße an Ampeln oder einfach Wetterverhältnisse überprüfen bzw. dokumentieren.

Da bei einer Verkehrsüberwachung allerdings eine Vielzahl solcher TMS zum Einsatz kommen, müssen die von den TMS erzeugten Daten an zentraler Stelle gesammelt, evaluiert und gemäß der rechtlichen Grundlagen zur Anstrengung von Verfahren gegen die Verkehrssünder genutzt werden. Die große Anzahl von TMS ergibt sich dabei durch die große Anzahl sinnvoller bzw. möglicher Messstellen im Straßenverkehrsnetz, da die TMS eher für einen dauerhaften Einsatz an einem Ort gedacht sind. Ferner müssen die erhobenen Beweisdaten gemäß der rechtlichen Grundlagen für eine Dauer von bis zu zehn Jahren (vgl. Abschnitt 2.7 (Seite 41)) beweiskräftig archiviert werden. Aus diesem Grunde wird in dem Konzept eine Archiving and Evaluation Unit (AEU) beschrieben, welche die genannten Aufgaben hat. Diese dient zudem dazu, die Sicherheit weiter zu erhöhen, da, wie [Mülo8, S.74] beschreibt, "eine zuverlässige Bewertung [..] nur auf einem separaten System erfolgen [kann, K.T.H.]."

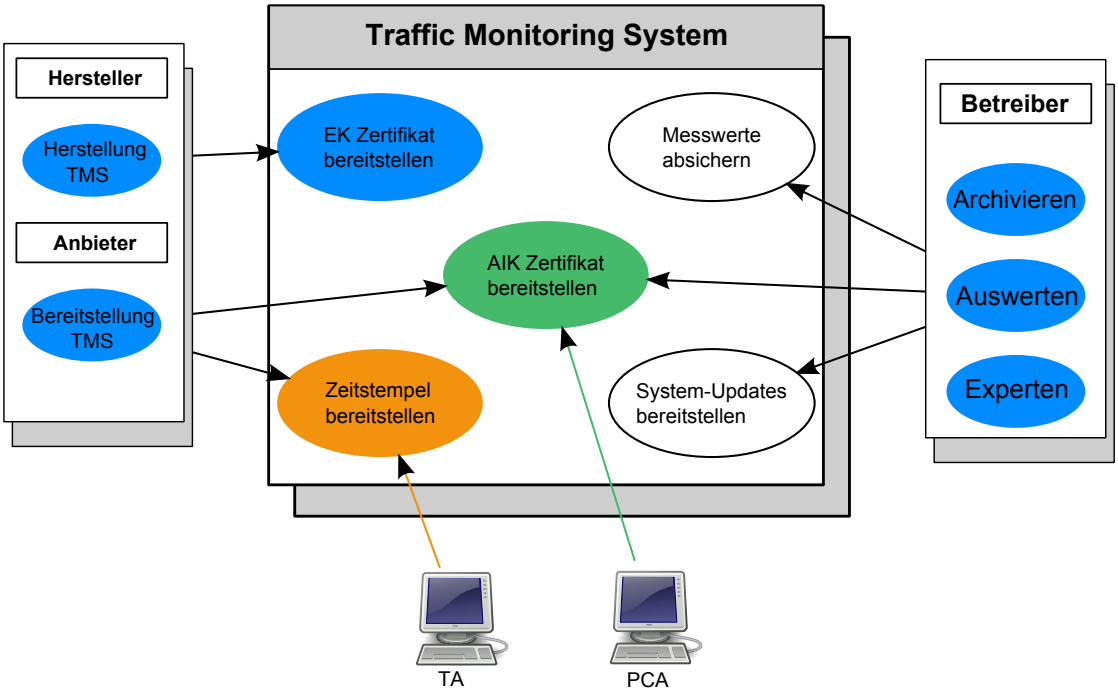

Abbildung 3.1 Rollen und Aufgaben in Anlehnung an [Ric09, Abb. 7]

In diesem Verwendungskontext gibt es verschiedene Rollen. Diese werden in Abbildung 3.1 gezeigt. Darin wird ein TMS zunächst von einem Hersteller gemäß dessen Prozessen gefertigt. Eine weitere Rolle ist der Anbieter der Verkehrsüberwachungslösung, welcher Service- bzw. Wartungsdienstleistungen für das Gesamtsystem anbietet. (vgl. [Ric09, Abb.6]) Außerdem gibt es eine weitere Instanz, welche die AEU bzw. TMS betreibt bzw. nutzt (vgl. *Betreiber* Abbildung 3.1). Des Weiteren wird eine so genannte Timing Authority (TA) benötigt, um eine vertrauenswürdige zeitliche Zuordnung von Ereignissen zu den ermittelten digitalen Daten durch eine Zuordnung des TPMs-internen Zeitzählers (vgl. Abschnitt 2.4.6.3 (Seite 33)) zu einer realen Zeit zu erstellen. Um den nötigen Datenschutz gewährleisten zu können, ist zudem eine so genannte PCA involviert, um die Identität des jeweiligen TMS zu pseudonomisieren, diese jedoch im Falle einer Gerichtsverhandlung auf richterliche Anordnung preisgeben zu können.

Dabei gibt es zu diesem Anwendungszweck einige Rechtsvorschriften, die in Abschnitt 2.7 (Seite 41) beschrieben worden sind. Daraus ergibt sich, dass ein Datensatz, welcher ein reales Ereignis in digitaler Form rechtssicher repräsentieren soll, im Folgenden Mea-

surement Record (MR) genannt, zumindest die Messwerte, im Folgenden Measurement Record Value (MRV) genannt, ein Datum und einen Ort enthalten muss. Einige denkbare MRVs sind außerdem z. B.:

- digitale Nachtsichtfotos (Infrarot),
- digitale Fotos,
- Zeiten,
- Fahrzeuggeschwindigkeiten,
- Fahrzeuganzahl,
- Fahrzeugabstände,
- Fahrzeugart und
- Schadstoffemissionswerte

(vgl. [Ric09, S.37]).

Welche MRVs genau benötigt werden, hängt von dem konkreten Einsatzzweck eines TMS ab. Als mögliche Einsatzszenarien werden von Richter sowohl Mautsysteme, Geschwindigkeitsmessungen als auch Abstandsmessungen oder generelle Verkehrsüberwachung zur Verhinderung bzw. Erkennung von Staus genannt. [Ric09, S.38] Ferner ist zum Beispiel in Städten eine Schadstoffemissionsüberwachung denkbar, um bei zu hoher Schadstoffkonzentration über Verkehrsleitbrücken das Geschwindigkeitslimit herabzusetzen oder um Schadstoffsünder in Umweltzonen zu ermitteln. Auch ein verlässliches Verkehrsleitsystem bzw. Verkehrsmanagement oder die Bevorzugung von öffentlichem Nahverkehr oder Rettungs- bzw. Polizeifahrzeugen ist denkbar. Zudem könnten Geschwindigkeitsübertretungen ggf. auch dann ermittelt werden, wenn der Fahrer, gewarnt vor dem „Blitzer“, jeweils bei den TMS bremst, da die Zeitdifferenz zwischen zwei TMS auf einer Strecke ohne Abzweigungen aufgrund der bekannten Distanz als Geschwindigkeitsmaß dienen kann (vgl. [Mlo09]).

3.1.1 Schutzziele

Wie bereits erwähnt, zielt das Konzept [Ric09] darauf ab, die Nichtabstreitbarkeit (vgl. Abschnitt 2.1 (Seite 8)) zu gewährleisten. Dabei wird in [Ric09, S.69,70] die Nichtabstreitbarkeit in diesem Falle so interpretiert, dass unverwechselbar eine Person identifizierbar ist, welche unabstreitbar gemäß der MRVs gehandelt hat. Um diese Nichtabsteitbarkeit dabei gewährleisten zu können, werden dem MR die folgenden zusätzlichen (zu den MRVs) sicherheitsrelevanten Informationen hinzugefügt:

- die Identität des TMS (ggf. nur implizit über die Identität des verwendeten TPMs - vgl. [Ric09, S.80]),
- die (Lauf-) Nummer des MR,
- der Standort des TMS (ggf. nur implizit über bekannten Aufstellungsort mit vorhandener Zuordnung zur Identität - vgl. [Ric09, S.80]),
- der Aufnahme-/Messzeitpunkt und
- die Systemkonfiguration bzw. -zustand zum Zeitpunkt der Messung

(vgl. [Ric09, S.70]).

Die Vollständigkeit der bei der AEU eingegangenen MRs bzw. MRVs ist sowohl durch die Nummern der MRs als auch durch den Systemzustandswert ermittelbar. Um die Integrität und Authentizität (vgl. Abschnitt 2.1 (Seite 8)) der MRs sicherzustellen, werden MRs zudem nach der Erstellung mit einer Signatur des TPM versehen bzw. abgesichert.

3.1.2 Annahmen

Im zu Grunde liegenden Konzept von Richter bestehen für die unterschiedlichen Sicherheitsprotokolle bzw. das Gesamtkonstrukt einige grundlegende Sicherheitsannahmen. (vgl.[Ric09, S.70]) Diese werden im Folgenden kurz aufgelistet bzw. erläutert. Dabei wird beispielsweise davon ausgegangen, dass die entsprechenden TPM-Fertigungsprozesse des Herstellers sicher sind bzw. in einer sicheren Umgebung durchgeführt werden. Es

wird auch angenommen, dass die ggf. entfernt angeschlossenen bzw. integrierten Mess-Sensoren korrekte, integere und authentische Werte liefern und den nötigen Datenschutz gewährleisten. Ferner wird gefordert, dass die AEU über ausreichend Speicherplatz verfügt, um die entstehenden Daten für eine Dauer gemäß den rechtlichen Vorgaben zu speichern. Außerdem wird eine korrekte und sichere Speicherung auf Seiten der AEU angenommen, sodass die MRs dort nicht später beschädigt bzw. verfälscht werden können. Zudem wird davon ausgegangen, dass die AEU, die TA und die PCA nicht kompromittiert worden sind und alle jeweils für sie geltenden Sicherheitsvorschriften einhalten. Letztlich wird für die Übertragung der MRs von den TMS zur AEU angenommen, dass diese ausreichend gemäß der rechtlichen Vorschriften gesichert ist.

Durch diese Annahmen wird der Blickwinkel des Konzeptes im Wesentlichen auf das vertrauenswürdige Hinzufügen von zusätzlichen Informationen (vgl. Abschnitt 3.1.1 (Seite 48)) zu den MRs bzw. die Sicherung der Inhalte der MRs beschränkt. (vgl. [Ric09, S.70])

Das Konzept kann außerdem lediglich gewährleisten, dass anhand der in den MRs dokumentierten Informationen erkennbar ist, wenn von einem TMS falsche bzw. ungültige MRVs bzw. MRs empfangen werden. Eine garantierte und korrekte Ermittlung, Verarbeitung und Übertragung von MRVs und MRs kann durch das Konzept jedoch nicht sichergestellt werden. (vgl.[Ric09, S.71])

3.1.3 Bedrohungen und Anforderungen

Bei der Erzeugung, Übertragung zur AEU und Auswertung von MRs unterliegen die AEU und die TMS verschiedenen Gefährdungen, welche eine anschließende rechtlich unbedenkliche Nutzung der MRs verhindern können. Eine umfassende Analyse der Schutzbedarfe, der im Konzept verwendeten Komponenten, gemäß den Grundschutz-Katalogen des Bundesamtes für Sicherheit in der Informationstechnik [BSI09], würde den Rahmen dieser und Richters Arbeit sprengen. Daher werden im zu Grunde liegenden Konzept [Ric09] lediglich ein paar intuitive zentrale Angriffsvektoren bzw. mögliche Schwachstel-

len (Vulnerabilities: $V_1, V_2, .., V_n$) diskutiert (vgl.[Ric09, S.41-48, T1-T9]). Diese werden im Folgenden, soweit möglich, zusammengefasst.

V_1 - **Bootprozess**

Bei der Betrachtung des Ressourcen-beschränkten TMS ist klar, dass eine Veränderung im Bootprozess, beispielsweise durch Austauschen von Teilen des Betriebssystems, die Sicherheit bzw. Glaubwürdigkeit des Systems erheblich gefährden würde, da die Anwendung bzw. Anwendungen zur Erfassung und Verarbeitung der Messergebnisse auf dieses aufsetzen. Zudem sind durch eine Kompromittierung von Teilen der in Abschnitt 2.4.3 (Seite 16) beschriebenen Vertrauenskette alle nachfolgenden Komponenten nicht länger vertrauenswürdig, da die Vertrauenskette unterbrochen wurde. Integritätswerte zur Plattformkonfiguration wären damit leicht vom unterliegenden System fälschbar und somit eine Kompromittierung bzw. abweichende Konfiguration ggf. nicht erkennbar (vgl. [Ric09, S.42, T1]). Die Nichtabstreitbarkeit, in Form der Authentizität und Integrität, und nicht zuletzt der Datenschutz wären damit vom TMS nicht mehr gewährleistet.

V_2 - **Exploit**

Ein weiterer Punkt ist das Einbringen von Schadcode in das TMS zur Laufzeit dessen. Wenn der Schadcode dabei in der Lage ist, beispielsweise durch Fehler in den Kommunikationsimplementierungen des TMS, Anwendungsprogramme oder Teile des Betriebssystems zu verändern bzw. zu ersetzen, wäre die Folge ähnlich dem oben beschriebenen *T1* ein Verlust der Nichtabstreitbarkeit und des Datenschutzes (vgl. [Ric09, S.43, T2]).

V_3 - **Rechteausweitung**

Wenn ein Angreifer in der Lage ist, auf dem TMS Rechte zu erhalten, beispielsweise unter Ausnutzung von Schwächen in den verwendeten Authentifizierungs- bzw. Autorisierungsverfahren, oder diese zu privilegierten Rechten auszuweiten, so sind die bereits genannten Schutzziele ebenso gefährdet, da mit den privilegierten Rechten der Austausch von Teilen der Systemsoftware oder ein direkter Zugriff auf MRVs oder kryptografische Schlüssel möglich wäre (vgl. [Ric09, S.44, T3+T4])

V_4 - **Man-in-the-middle-Angriff**

Wenn es dem Angreifer gelingt, durch Fehler im Authentifizierungsprozess, dem

TMS vorzutäuschen, dass dieses mit dem korrekten Kommunikationspartner kommuniziert, beispielsweise der AEU, so ist er in der Lage, im Namen dieser durch Softwareupdates die Betriebssoftware des TMS zu verändern. Ferner ist denkbar, dass dieser, in Kombination mit dem in *Rechteausweitung* genannten Zugriff auf Schlüsselinformationen, sich das TMS als eigene Überwachungseinheit nutzbar macht oder sich selbst gegenüber der AEU als eines der erfolgreich angegriffenen TMS präsentiert (vgl. [Ric09, S.45,T5+T6]). Außerdem ist ein reiner *Man-in-the-middle*-Angriff vorstellbar, bei dem der Angreifer lediglich Teile der zwischen dem TMS und der AEU ausgetauschten Nachrichten verändert, um beispielsweise die Höhe der Geschwindigkeitsübertretung zu verringern. Zu diesem Zweck gibt sich der Angreifer gegenüber der AEU als das TMS aus und gegenüber dem TMS als AEU.

V_5 - **Denial-of-Service-Angriff**

Ein Angreifer, welcher durch Überflutung bzw. verschiedene andere Methoden erreicht, dass keine vollständigen bzw. korrekten MR mehr erzeugt werden können, ist in der Lage, ggf. sogar Messungen des TMS vor bzw. nach einem entsprechenden Angriff unbrauchbar zu machen, da diese einer statistischen Überprüfung u. U. nicht mehr standhalten. Ein solcher Angriff ist beispielsweise auf die ggf. externen oder auch internen Sensoren des TMS denkbar (vgl. [Ric09, S.46,T7]).

Richter nennt hier (vgl. [Ric09, S.46]) beispielsweise Systeme für Verkehrsteilnehmer, welche durch elektromagnetische Impulse Fehlfunktionen bzw. fehlerhafte Messwerte in Messgeräten auslösen. Diese sind jedoch, wie in Abschnitt 2.7 (Seite 41) beschrieben, zumindest in Deutschland per Gesetz verboten.

V_6 - **interne Fehler**

Ein Risiko geht zudem davon aus, dass die innerhalb des TMS verwendeten Hardware- oder Softwarekomponenten Fehler, wie beispielsweise Pufferüberläufe oder Berechnungsfehler, aufweisen, da dies ein Fehlverhalten des TMS auslösen kann. Auch hier ist dadurch die Nichtabstreitbarkeit gefährdet, da das TMS nicht unwiderlegbar korrekt arbeitet. (vgl. [Ric09, S.47,T8])

V_7 - **äußere Einflüsse**

Ein weiterer Risikofaktor sind äußere Einflüsse, wie beispielsweise Witterungseinflüsse oder Vandalismus, da diese ggf., wie in *Denial-of-Service* beschrieben, zu ei-

nem temporären, teilweisen oder dauerhaften Ausfall von Komponenten des TMS führen können und ggf. dabei, wie bereits beschrieben, ungültige oder keine Daten mehr erzeugt werden.(vgl. [Ric09, S.47,T9])

Aus den erkannten Schwachstellen ($V_1, .., V_7$ bzw. T1-T9 aus Richters Arbeit [Ric09, S.41-48]) lassen sich eine Reihe von Anforderungen bzw. benötigten Gegenmaßnahmen (Solutions: $S_1, S_2, .., S_n$) extrahieren. (vgl. [Ric09, S.48-57,Abschnitt 4.2.2,R1-R11]) Diese werden nun im Folgenden kurz zusammengefasst aufgelistet.

S_1 - **sicherer Bootprozess**

Da eine gewollte oder ungewollte Veränderung des Bootprozesses des TMS die Vertrauenswürdigkeit dessen, wie in V_1 beschrieben, erheblich beeinträchtigt, muss ein sicherer Bootprozess realisiert werden. Bei diesem muss zumindest erkannt werden, dass sich das TMS nach einem veränderten Bootprozess in einem anderen Zustand befindet als bei einem normalen Startvorgang. Um dies realisieren zu können, wird im vorliegenden Konzept auf das TCG-Konzept des „trusted boot“ zurückgegriffen, welches beim Startvorgang, noch vor dem Betriebssystemstart, die Konfiguration bzw. Identifikation jeder Systemkomponente in Form eines Hashwertes in ein PCR hinzufügt. Der Wert dieses PCR kann später mit einem vorliegenden Referenzwert verglichen werden, sodass die in diesem Systemzustand erzeugten Messwerte ggf. nicht fälschlicherweise als glaubwürdig interpretiert werden.

Durch die Sicherstellung der Integrität der beim Systemstart verwendeten Komponenten, können hiermit auch V_6 bzw. V_7 adressiert werden, da ein Ausfall von Komponenten in Folge von Vandalismus oder Witterungseinflüssen zumindest bei einem erneuten Start des TMS erkennbar wäre. Auch ist damit eine Veränderung von Softwarekomponenten bzw. Hardwarekomponenten, welche durch fehlerhafte Hard- oder Software verursacht wurde, beim Systemstart dokumentierbar. (vgl. [Ric09, S.49,50])

S_2 - **Attestierung der Systemkonfiguration**

Da eine Veränderung der Anwendungssoftware auf dem TMS einen Verlust der Vertrauenswürdigkeit bedeuten würde (vgl. V_2), wird eine Möglichkeit benötigt, zu einem späteren Zeitpunkt nachzuweisen, dass das TMS sich zum Zeitpunkt einer Messung in einem zulässigen Zustand bzw. einer zulässigen Konfiguration befun-

den hat. Um dies zu erreichen, wird im zu Grunde liegenden Konzept das TCG-Konzept der Plattform-Attestierung (vgl. Abschnitt 2.4.6.4 (Seite 34)) eingesetzt. Dabei werden Prüfsummen in Form von Hashwerten über verwendete Anwendungen und Einstellungen erstellt und mit bekannten Hashwerten verglichen bevor ein Kommando dieser ausgeführt wird. (vgl. [Ric09, S.50-51,R2+R3+R4])

S_3 - **Sichere Erstellung, Verarbeitung, Verwendung und Speicherung von Daten**

Wird ein Zugriff auf vertrauenswürdige Messdaten oder Schlüsselmaterial ohne Authentifizierung und Autorisierung ermöglicht oder werden diese beschädigt, so sind diese bzw. damit beglaubigte Daten nicht länger vertrauenswürdig. Aus diesem Grunde wird eine sichere Erstellung, Verarbeitung und Speicherung der Daten bzw. des Schlüsselmaterials im TMS benötigt. Zu diesem Zweck werden im TMS die Möglichkeiten des TPM zur sicheren Speicherung und Erstellung von Schlüsseln genutzt, um mithilfe dieser die Nutzdaten verschlüsseln bzw. signieren und so die Authentizität bzw. Integrität der Daten gewährleisten zu können. Ein nicht autorisierter Zugriff auf den Inhalt der Daten wird durch die dafür nötige Authentifizierung am TPM und dessen Autorisierung unterbunden. Zusätzlich wird durch den Einsatz von System Management Logs (SMLs) durch den TSS gewährleistet, dass jede Veränderung der Daten dokumentiert wird. Zudem wird eine Löschung von Daten durch eine Beschränkung von Systemoperationen vom Betriebssystem verhindert, wenn keine gültige Autorisierung durch das TPM vorliegt, und im SML geloggt. Mit den genannten Maßnahmen werden dabei die Schwachstellen $V_2 - V_4$ adressiert, da eine sichere Authentifizierung bzw. Autorisierung durch das TPM und zusätzlich dessen Vertrauenswürdigkeit über das SML abgesichert wird (vgl. [Ric09, S.52-53,R5+R6])

S_4 - **Zugriffsbeschränkungen**

Wie in S_3 bereits beschrieben, ist die Vertrauenswürdigkeit des gesamten Systems gefährdet, wenn keine Zugriffsrechte erforderlich sind, um Daten zu lesen, zu verändern oder zu löschen. Daher wird dies über kryptografische Schlüssel abgesichert. Dabei bekommen nur Identitäten entsprechende Zugriffsrechte, die nachweisen können, dass sie den jeweiligen Schlüssel besitzen. Für das Schlüsselmanagement wird das TPM in Verbindung mit dem TSS genutzt. Zudem können mittels

Zertifikaten verschiedene Zugriffsrechte an eine Identität und einen Systemzustand gebunden werden. Wie in S_3 beschrieben, übernimmt damit das TPM die Authentifizierung und die Autorisierung. Durch die Verwendung des TPM zur Authentifikation, Autorisierung und zum Schlüsselmanagement und der Annahme, dass dieses sicher ist, wird so die Schwachstelle V_3 adressiert, da eine Anmeldung am TPM damit abgesichert ist. (vgl. [Ric09, S.54,R7])

S_5 - **sichere Identifikation**

Wenn die sichere Identifikation eines TMS oder einer zulässigen AEU nicht sichergestellt ist, so ist die Vertrauenswürdigkeit des Systems nicht mehr gewährleistet, da (vgl. V_4) eine Kompromittierung des TMS bereits durch vermeintlich korrekte Softwareupdates möglich ist, Daten unbemerkt von Dritten gesammelt oder als vermeintliches TMS eingespeist werden können. Um dies zu verhindern, wird im Konzept die eindeutige Identität des TPM in Form eines EK (vgl. Abschnitt 2.4.3 (Seite 16)) als Identität für das TMS verwendet. Dabei werden zudem die Möglichkeiten des TPM bzw. TSS genutzt, durch Zertifikate bzw. Signaturen auf Basis von nicht-migrierbaren Signaturschlüsseln verifizierbare und authentische Nachrichten zu erzeugen. In Verbindung mit sicheren Authentifizierungs- und Verschlüsselungsprotokollen wird dadurch eine sichere Verschlüsselung der Kommunikation ermöglicht, sodass ausschließlich die korrekte AEU die Daten entschlüsseln und somit nutzen und aufgrund der Signatur des TPM des TMS einem konkreten TMS zuordnen kann. Den mit der Signatur geschützten Daten werden außerdem Prüfsummen hinzugefügt um die Integrität bzw. mit fortlaufenden Zählern bzw. Hashwerten bzw. die Vollständigkeit der übertragenen MR zu sichern. So kann eine AEU einen fehlenden MR sofort anhand der falschen Hashwerte entdecken. Die zusätzliche Ergänzung der MR um qualifizierte Zeitstempel sichert zudem den Zeitpunkt des korrespondierenden realen Ereignisses. Dies adressiert damit auch V_5, da entsprechende *Denial-of-Service-Angriffe* zumindest von der AEU auf statistischer Basis erkannt werden können, die bereits erzeugten Daten entsprechend gesichert sind, um ihre Vertrauenswürdigkeit zu erhalten, und keine MR erzeugt werden, wenn sich das TMS nicht in einem zulässigen Zustand befindet.

S_6 - **sichere Hardware**

Das TMS wird gegen die in V_7 genannten äußeren Einflüsse durch den Einsatz entsprechend zertifizierter bzw. gehärteter Gehäuse abgesichert. Diese werden zudem versiegelt, um eine Öffnung bzw. Beschädigung des TMS eindeutig erkennen zu können. [Ric09, S.56,R10]

Denkbar ist außerdem, dass dem TPM bzw. TMS der Status dieser Versiegelungen zugänglich gemacht wird, um die Attestierung der Plattform um diesen Status zu erweitern. Auch kann dieser Status genutzt werden, um für die Erzeugung vertrauenswürdiger MRs notwendige Schlüssel an eine Systemkonfiguration zu binden, bei der die Versiegelung intakt ist.

3.1.4 Ablauf

Die Interaktion der in Abschnitt 3.1.3 (Seite 49) genannten Lösungen wird im Folgenden in einem logischen Ablauf des Konzeptes [Ric09] beschrieben. Dabei werden die in Abschnitt 3.1 (Seite 44) genannten Rollen und Entitäten verwendet und ggf. näher beschrieben.

Der Ablauf gliedert sich in verschiedene logische Teilabläufe, die nur einmalig bei der Herstellung eines TMS oder dauerhaft bei jeder Messung eines MRs durchlaufen werden müssen. Daher folgt zunächst die Beschreibung eines groben Herstellungsprozesses in Abschnitt 3.1.4.1. Im Anschluss folgt eine Beschreibung des Bereitstellungsprozesses in Abschnitt 3.1.4.2 (Seite 56), in dem die so hergestellten TMS auf ihre reale Verwendung vorbereitet werden. Letztlich wird ein beispielhafter Erfassungsprozess von Verkehrssündern in Abschnitt 3.1.4.3 (Seite 57) beschrieben.

3.1.4.1 Herstellung

Bei der Herstellung der TMS wird aufgrund der von Richter im Konzept verankerten Nutzung von TPMs zunächst ein TPM hergestellt. Für das neu erstellte TPM wird vom

Hersteller im TPM ein EK generiert und im nicht flüchtigen Speicher des TPM gespeichert (vgl. Abschnitt 2.4.3 (Seite 16) bzw. Abschnitt 2.4.4 (Seite 24)), um die in S_5 geschilderte *sichere Identifikation* zu gewährleisten. (vgl. [Mülo8, S.34],[Rico9, S.39]) Zusätzlich wird ein EK-Zertifikat (EKCRED) angefertigt, welches bescheinigt, dass der EK bzw. das TPM gemäß dem vom Hersteller dafür definierten Prozess erzeugt wurde. Das so erstellte EKCRED bildet mit Conformance- (CCRED) und Validation-Zertifikaten (VCRED) die so genannten und ggf. im TPM gespeicherten Plattform-Zertifikate (PCRED vgl. [Mülo8, S.61]). Das CCRED bestätigt, dass das TPM gemäß der TCG-Spezifikation korrekt entwickelt und implementiert wurde.

Im nächsten Schritt wird das TPM dann in die Rechnerplattform, die die Grundlage für das TMS bilden soll, gemäß den TCG-Spezifikationen (vgl. [TCG11c, S.87]) integriert. Danach wird das CCRED erstellt und entweder im nicht flüchtigen Teil des TPM gespeichert oder auf anderem Wege dem Anbieter zur Verfügung gestellt. Beispielsweise können diese, wie in Abbildung B.1 (Seite 96) angedeutet, mit einem *Storage Key* (vgl. Abschnitt 2.4.3 (Seite 16)) verschlüsselt extern gespeichert werden.

Es folgt die Erstellung und Zertifizierung einer Referenzmessung der Betriebssoftware als Basis für die in S_2 genannte *Attestierung der Systemkonfiguration* bzw. den in S_1 genannten *sicheren Bootprozess*. Diese wird in Form von VCREDs ebenfalls entweder im TPM gespeichert oder auf anderem Wege (s.o.) bereitgestellt.

Im letzten Schritt der Herstellung wird nach positiv erfolgtem Funktionstest die eingebettete TMS-Plattform in ein Gehäuse integriert, welches die in S_6 genannten Eigenschaften erfüllt. Die Versiegelung des Gehäuses schließt diesen Prozess ab.

Optional können bereits beim Hersteller die für den Betrieb nötigen AIKs bzw. Signaturschlüssel erstellt und dem Betreiber für dessen AEU die zugehörigen AICs übermittelt werden.

3.1.4.2 Bereitstellung

Die Bereitstellung der TMS obliegt nach Richter dem Anbieter [Rico9, S.39]. Das vom Hersteller erzeugte neue TMS, welches mit den PCREDs ausgeliefert wird, wird vom An-

bieter zunächst an der vom Betreiber geforderten Stelle gemäß den rechtlichen Grundlagen in Abschnitt 2.7 (Seite 41) installiert. Danach wird eine initiale Verbindung zur AEU des Betreibers oder Anbieters hergestellt und diese mit allen notwendigen Metadaten, beispielsweise dem genauen Ort und der an das TMS vergebenen ID, versorgt. Im Anschluss daran werden die vorgeschriebenen Referenzmessungen durchgeführt und in Form von MRs gespeichert und zur AEU übertragen bzw. vor Ort erfasst und dokumentiert. Da diese Daten zur genauen Beschreibung der Platzierung bzw. Identifikation des TMS dienen, wird damit die Grundlage für S_2 und S_5 gelegt.

3.1.4.3 Nutzung

Die Nutzung der TMS kann zunächst grob in zwei verschiedene Prozesse unterteilt werden. So muss vor der *Erfassung*, wenn ein TMS neu gestartet wurde, eine *Initialisierung* durchgeführt werden, bevor ein TMS in der Lage ist, gemäß den rechtlichen Grundlagen, Verkehrsverstöße rechtssicher zu *erfassen*.

Initialisierung

Bei der *Initialisierung* wird, wie in S_1 beschrieben, zunächst ein sicherer Systemstart durchgeführt, um auf Basis dessen zu einem sicheren Betriebsstatus zu gelangen. Dieser wird dann durch die in S_2 beschriebene *Attestierung der Systemkonfiguration* von einer PCA beglaubigt. Mit den so erzeugten AIK bzw. AIC wird dann zunächst ein neu erzeugter Signaturschlüssel signiert und dann ein Synchronisationsprozess angestoßen, um den TPM-internen Zähler an eine reale und sichere Zeitangabe kryptografisch zu binden. Hierfür wird von Richter eine von Challenger et. al bzw. in der Spezifikation (vgl. [Cha+07, S.277], [TCG11d, S.104-109]) vorgeschlagenes Protokoll zur Zeitsynchronisation mit einer so genannten *Time Stamping Authority* bzw. TA als so genanntes *TA-Protokoll* verwendet. [Ric09, Abschnitt 4.6.3]

Dabei wird zunächst eine Nonce mithilfe der TCG Service Provider Interface (TSPI)-Methode „Tspi_TPM_TickStampBlob“ an den aktuellen Stand des TPM-internen Zeit-

zählers und das TPM gebunden. Der Zählerstand stellt die vergangenen Einheiten seit dem Beginn einer Zeitsitzung im TPM dar. Daher ist die Nonce nun an einen spezifischen Zählerstand einer konkreten Zeitsitzung eines TPM gebunden.

Da in der Spezifikation [TCG11d, S.104-108] jedoch nicht konkret festgelegt wurde, wie ein TPM in unterschiedlichen Energiesituationen mit dem aktuellen Zeitzähler verfährt und auch keine dauerhafte Versorgung des TPM mit Strom gefordert wird, ist eine direkte Verwendung des TPM-internen Zeitzählers im besten Fall Hersteller- und Plattformabhängig. Da dies für eine verlässliche und vertrauenswürdige Zeitangabe in den MRs nicht ausreichend ist, wird bei jeder kritischen Änderung an der Energieversorgung und in regelmäßigen Abständen eine neue Zeitsitzung gestartet, um eine möglichst geringe Abweichung oder wieder eine valide Zuordnung der jeweils aktuellen *Timing Session* zu haben. Zudem wird eine Zeitsitzung stets mithilfe einer entfernten TA einem realen Zeitintervall zugeordnet, um eine durchgängige und verlässliche Zeitangabe für das Auftreten der realen Ereignisse, welche vom TMS erfasst werden, zu erhalten.

Eine direkte Zuordnung eines Zählerstandes zu einem realen Zeitstempel, bei der entfernten TA, könnte aufgrund von Verzögerungen bei der Übertragung zu erheblichen Fehlern führen. Aus diesem Grund ist nur eine Bestimmung eines Zeitintervalls möglich. Um dies zu erreichen, wird das Ergebnis der „Tspi_TPM_TickStampBlob“-Methode zur TA übertragen.

Diese prüft die Identität bzw. Echtheit des TPM anhand eines AIC und signiert dann einen realen Zeitstempel (*Timestamp*) zusammen mit den übertragenen Daten und schickt dies, mit dem öffentlichen AIK-Schlüssel verschlüsselt, zurück zum TPM. Durch die Verschlüsselung mit dem öffentlichen Teil des AIK ist nur das spezifische TPM in der Lage, diese zu entschlüsseln. (vgl. Abschnitt 2.4.3 (Seite 16))

Im letzten Schritt entschlüsselt das TPM den Zeitstempel bzw. die Signatur der TA und bindet diese wiederum mittels „Tspi_TPM_TickStampBlob“ an einen nun aktuellen Zählerstand. Damit lässt sich der reale Zeitstempel der TA eindeutig dem Intervall zwischen dem ersten und zweiten TickStampBlob zuordnen.

Erfassung

Die *Erfassung* besteht zunächst aus der Sammlung der verschiedenen MRVs von den angeschlossenen oder in das TMS integrierten Messsensoren. Die so erfassten MRVs werden dann in einen MR integriert und um die in Abschnitt 3.1.1 (Seite 48) erwähnten zusätzlichen Informationen ergänzt. Um den Zeitpunkt dabei zu ergänzen, wird zusätzlich zum aktuellen Wert des TPM-Zählers die kryptografische Zuordnung der aktuellen Timing-Session an eine reale Zeit der TA in Form von Signaturen in den MR integriert. Auch eine vertrauenswürdige Systemkonfiguration wird nach Richter explizit über die Integration von PCR-Werten in den MR gewährleistet. Zudem wird der Signaturschlüssel, mit dem die MRs signiert werden, an die bei der Herstellung im Rahmen der Referenzmessung ermittelten PCR-Werte gebunden. (vgl. Abschnitt 2.4.3 (Seite 16)) Letztlich wird der neu erzeugte MR durch das TPM mit dessen dafür erzeugten Signaturschlüssel signiert und die Signatur dem MR hinzugefügt. Der erzeugte Signaturschlüssel wurde dabei selbst zuvor durch einen AIK signiert, sodass die Authentizität der MRs gewährleistet wird, wenn der AEU das zugehörige AIC bekannt ist. Im letzten Schritt werden neu erzeugte MRs zur zugehörigen AEU übertragen und dort geprüft, gespeichert und ggf. zur Ermittlung von Verkehrssündern genutzt. Die dabei gemäß der Fokussierung des Konzeptes optionale Verschlüsselung wird im Konzept von Richter als gegeben angenommen, da sich das Konzept auf die Absicherung der MRVs durch das Hinzufügen von zusätzlichen Informationen gemäß den TCG-Konzepten beschränkt.(vgl. Abschnitt 3.1.2 (Seite 48) bzw. [Ric09, S.70,71])

3.2 Widersprüche/Unklarheiten

Wie bereits in Abschnitt 1.3 (Seite 5) und Abschnitt 3.1 (Seite 44) erwähnt, gab es neben der Komplexität des *Trusted Computings* bei der Analyse des zu Grunde liegenden Konzeptes von Richter auch einige Unklarheiten oder Aussagen, die einander widersprachen. Im folgenden Abschnitt werden einige Beispiele aufgezeigt, um einen Überblick über die

Fehleranfälligkeit bzw. Übersichtlichkeit von Konzepten rund um das *Trusted Computing* aufgrund der enormen Komplexität geben zu können.

Manche leicht zu erkennende bzw. zu findende Fehler sind vergleichsweise einfach zu verstehen. Beispielsweise wird in Richters Arbeit von der Signatur von Daten mit einem öffentlichen Schlüsselteil eines asymmetrischen Schlüsselpaares, gesprochen : "Another important use of the public portion of the EK is to sign Attestation Identity Keys (AIKs) so that they are bound to one specific TPM." [Ric09, S.26] Dies ist offensichtlich nicht korrekt, da beim Signieren normalerweise ein **privater Schlüssel** zum *Verschlüsseln* der zu signierenden Daten verwendet wird (vgl. Abschnitt 2.3.3 (Seite 12)).

Allerdings finden sich auch Widersprüche, die über die normalen Grundlagen hinaus gehen. So beschreibt Richter den EK in [Ric09, Abschnitt 3.4.1] mit den Worten "The private portion of the EK should never leave the TPM", andererseits wird im Abschnitt über das Signieren von Daten [Ric09, S.87,88] das Gegenteil behauptet: "To receive an AIK credential from the PCA, the generated AIK and the private portion of the EK must be transmitted to the PCA". Die TCG-Spezifikation sorgt hier jedoch für Klarheit: "The PRIVEK MUST never be out of the control of a TPM shielded location".[TCG11d, S.32]

Für etwas mehr Unklarheit hingegen sorgen falsche bzw. widersprüchliche Aussagen zur Verwendung der so genannten *Credentials* in den von Richter beschriebenen TCG-Konzepten. Im Abschnitt "Attestation" [Ric09, S.29] schreibt Richter z.B. : "The EK and a specially for this session created AIK key are sent to the PCA." und auf der Seite davor "Endorsement Key (EK) is an asymmetric non-migratable key. [...]". Durch diese beiden Aussagen könnte der Verdacht entstehen, dass das komplette EK-Schlüsselpaar zur PCA geschickt werden soll. Dies wird allerdings bereits von dem Teil, dass es sich um ein nicht migrierbares Schlüsselpaar oder viel mehr einen nicht migrierbaren privaten Schlüssel handelt, widerlegt. Interessant ist zudem, dass der Aussage nach auch das ebenfalls nicht migrierbare AIK-Schlüsselpaar übertragen werden soll. Korrekterweise werden allerdings von beiden Schlüsselpaaren nur die öffentlichen Anteile übertragen und zudem bei der Übertragung mit dem öffentlichen Schlüssel der PCA verschlüsselt. Hinzu kommen außerdem noch, wie bereits in Abschnitt 2.4.6.4 (Seite 34) beschrieben, eine Menge von Zertifikaten bzw. *Credentials* um der PCA eine Prüfung der Echtheit des TPMs überhaupt zu ermöglichen.

Leider gibt es bei Richter auch einige missverständliche Aussagen. Die folgende Aussage : "**Real time** UTC time obtained from the Timing Authority. Ideally, this timing value is consistent with the event time value. But in general, will deviate from the event time value for some millisecond."[Ric09, S.96] von Richter könnte beispielsweise so interpretiert werden, dass bei jedem Event eine UTC-Zeit von der TA bezogen wird. Außerdem wird davon gesprochen, dass der von der TA erstellte Zeitstempel zurück zum TMS geschickt und dort dieser zu der vorher vom TMS erstellten Nonce hinzugefügt wird.[Ric09, S.96] Tatsächlich wird jedoch auf der folgenden Seite in Richters Konzept dargelegt, dass zunächst eine Nonce generiert, diese mittels der „Tspi_TPM_TickStampBlob"-Methode an den aktuellen Zählerstand des TPMs gebunden und dessen Ergebnis dann zur TA übertragen wird. Ferner erzeugt diese dann einen Zeitstempel über das Ergebnis des ersten Aufrufes der „Tspi_TPM_TickStampBlob"-Methode, überträgt diesen zurück zum TMS, bei dem widerum erneut durch die „Tspi_TPM_TickStampBlob"-Methode an einen weiteren Zählerstand des TPM gebunden wird.

Weiterhin wird im Implementierungsabschnitt von Richter davon gesprochen, dass der *TAClient* den öffentlichen Teil eines AIKs mit dem öffentlichen Schlüssel der TA verschlüsselt an diese sendet. Diese soll damit in der Lage sein, die Anfrage zu überprüfen, eine Nonce zu erzeugen und mit dem öffentlichen Teil des EK zurück zum TMS zu übertragen. [Ric09, S.115] Interessant ist hier einerseits, dass ein AIK keinen Verweis auf einen öffentlichen Teil des EK enthält und somit unklar ist, woher die TA diesen erhalten hat, und zum anderen, dass Richter bereits auf Seite 102 in einem beispielhaften Protokollablauf, entgegen der oben genannten Aussagen, beschrieben hat, dass ein AIC zur TA übertragen und dort überprüft werden soll. Ein AIC enthält (vgl. Abschnitt 2.4.3 (Seite 16)) den öffentlichen Schlüssel des AIK und zudem eine Signatur einer vertrauenswürdigen PCA, welche die Echtheit des TPMs bestätigt, ohne dessen eindeutige Identität in Form des öffentlichen Teils des EK preis zu geben. Allerdings wird dort auch nur davon gesprochen, dass eine Implementierung so ähnlich konzipiert sein könnte. Daher ist davon auszugehen, dass hier eher Bezug auf das *TA-Protocol* [Ric09, S.117 bzw. Abbildung 24] genommen wird. Diese Abbildung wurde, da in ihr mehrere fragwürdige Details enthalten sind, in diese Arbeit überführt und ist in Abbildung 3.2 (Seite 62) zu sehen.

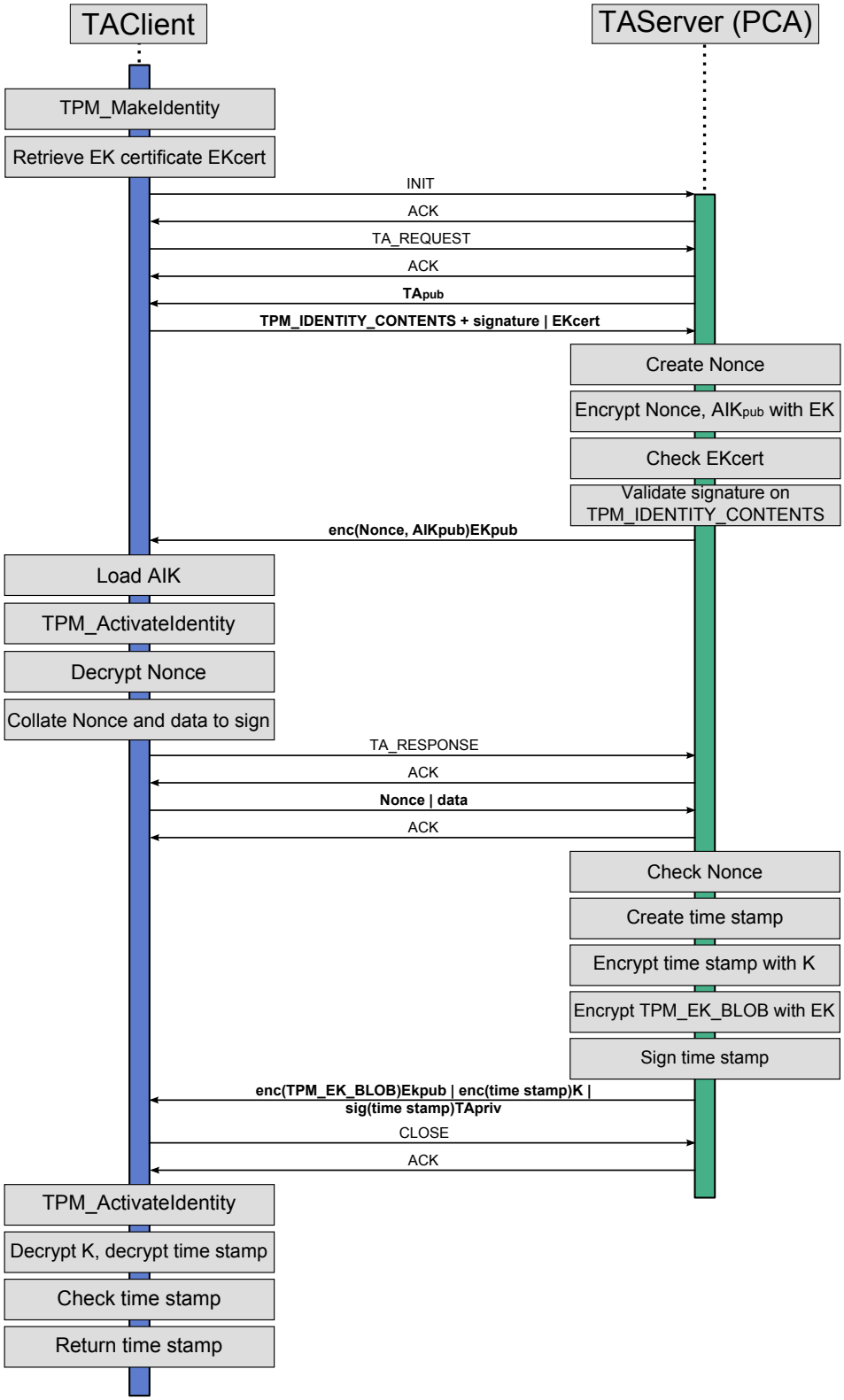

Abbildung 3.2 TA Protokoll nach Richter [Ric09, Abb. 24]

Im vorliegenden *TA-Protocol* nach Richter, welches in Abbildung 3.2 (Seite 62) zu sehen ist, wird es bereits beim Verständnis des ersten Nachrichtenaustausches vom TMS zur PCA problematisch. Um genauere Informationen zu erhalten, welche Komponenten *TPM_IDENTITY_CONTENTS* enthält bzw. welcher Signaturschlüssel hier verwendet wird, kann hinsichtlich der Signatur auf die Aussagen von Richter [Ric09, S.113-115] zurückgegriffen werden.

Bzgl. der genauen Inhalte der *TPM_IDENTITY_CONTENTS*-Datenstruktur lässt sich in den Spezifikationen der TCG Folgendes herausfinden : "TPM_MakeIdentity uses this structure and the signature of this structure goes to a privacy CA during the certification process." [TCG11e, Abschnitt 12.5]. Unterstrichen wird diese Aussage dabei auf der Mailinglist der Entwickler eines open-source TSS [Tro11] : "TCPA_IDENTITY_CONTENTS is the struct which is signed by the AIK in the request which is sent to the Privacy CA. However, the struct itself is not sent along (in fact it is never exported from the TPM)." [Fin06] Damit ist bereits im ersten Schritt des *TA-Protocols* eine deutliche Ungenauigkeit bzw. ggf. ein durch einen hohen Abstraktionsgrad bedingter Fehler enthalten. Dieser vermeintliche Fehler scheint sich allerdings in mindestens einer weiteren Publikation wiederzufinden : "The process of AIK certification is shown in figure 1. First, the client creates a new AIK using the TPM_MakeIdentity command. This creates the key and the TPM_IDENTITY_CONTENTS structure which contains the public AIK. This structure is then signed with the AIK and sent to the PCA together with the public EK and its certificate." [LKS09, S.5]. Ein kompletter Ablauf von der Erzeugung eines AIK im TPM bis hin zur Signierung des öffentlichen Schlüsselteils des AIK mit dem privaten Schlüsselteil des AIK ist in [TCG11a, Seite 10] zu finden. Zudem beschrieb auch Fichtinger dies bereits im Jahre 2007 korrekt. [Fic07, S.55].

Kapitel 4

Vereinfachte skalierbare Darstellung

Aufgrund der in Abschnitt 2.4 (Seite 13) beschriebenen Komplexität und den daraus resultierenden Irrtümern, Unklarheiten oder Widersprüchen, von denen in Abschnitt 3.2 (Seite 59) einige aufgeführt werden, wird eine Darstellungsform benötigt, die für Klarheit sorgt. In Abschnitt 4.1 wird daher eine vom Autor gewählte Form vorgestellt und anhand dieser werden verschiedene Aspekte aufgezeigt und erläutert. Zudem wird darauf eingegangen, weshalb der Autor der Meinung ist, dass diese Form der Modellierung leichter verständlich ist. Darauffolgend zeigt und erklärt Abschnitt 4.2 (Seite 68) ein paar exemplarische Auszüge des Konzeptes von Richter in der beschriebenen Notation.

4.1 Begründung

Im Verlaufe der Arbeit war als Darstellungsform eigentlich eine reine Formalisierung gemäß der in Abschnitt 2.6 (Seite 38) beschriebenen Notation exemplarisch für einige Teile geplant. Allerdings bietet dieser Ansatz ausschließlich eine Nachrichten-zentrierte Darstellung, die zudem nicht auf verschiedene Verständnis- bzw. Detailstufen skaliert werden kann.

Aus diesem Grund entschied sich der Autor für eine neue Darstellungsform auf Basis der Business Process Modelling Notation in Version 2.0 (BPMN 2.0). Hinzu kommt, dass für die BPMN 2.0 eine gute Werkzeugunterstützung verfügbar ist. So kann ein Anwender bereits aus mindestens drei freien Varianten, Yaoqiang BPMN Editor [Ble11], Bonita Open Solution [bon11] und BPMN2 Editor für Eclipse [ime11], wählen. Zudem gibt es mit Activiti [act11] mindestens eine freie Lösung, um in BPMN 2.0 modellierte Prozesse ausführen zu lassen. Die wichtigsten Komponenten, von denen einige die Skalierbarkeit der Notation unterstreichen, werden in Abschnitt 4.1.1 (Seite 67) kurz erläutert. In Abschnitt 4.1.2 (Seite 68) folgt dann ein kurzer Ausblick darüber, was mit dieser Notation in Zukunft alles möglich sein könnte bzw. wo sie thematisch ähnlich verwendet wird.

4.1.1 Komponenten

Diese Notation bietet aufgrund der Orientierung an den Anforderungen der Wirtschaft eine relativ beliebige Skalierung des Detailgrades einer Darstellung, da sich so genannte „Unter-Prozesse“ einsetzen lassen. (vgl. [All09, S.86]) Diese können vollständig oder komprimiert (eingeklappt) angezeigt werden. Dadurch lassen sich bereits sehr große Prozesse stark vereinfacht darstellen. Da dieses Verfahren jedoch auch rekursiv mehrfach verwendet kann, d.h. in einem „Unter-Prozess“ sind weitere „Unter-Prozesse“ eingesetzt, kann ein Ablauf, dessen detaillierterer Ablauf bekannt ist, beliebig abstrahiert und vereinfacht dargestellt werden.

Neben den Sequenzflüssen der Prozesse in Kolloborations-, Choreographie- oder Konversationsdiagrammen können auch Nachrichtenflüsse zwischen Prozessen von verschiedenen Entitäten dargestellt werden. (vgl. [All09, S.49, S.138, S.149]) Auch können Sequenzflüsse innerhalb eines Prozesses einer Entität in verschiedene Bereiche (so genannte “swimlanes”) untergliedert und durch auf- und zuklappen flexibel skaliert werden. Ein Prozess einer Entität ist zunächst in einem “pool” untergebracht und kann durch die “swimlanes” in mehrere logische Bereiche (im organisatorischen Bereich z.B. Abteilungen) unterteilt werden.(vgl. [All09, S.17])

Des Weiteren unterstützt die BPMN 2.0 umfangreiche Möglichkeiten bei der Modellierung von Ausnahme- und Fehlersituationen sowie ereignisbasierte Prozesse bzw. Prozessschritte. (vgl. [All09, S.63,S.107,S.111]) Es kann z.B. ein Prozess alle 15 Minuten mit einem zeitabhängigen Startknoten aktiviert werden. Es können zudem Teilschritte oder „Unter-Prozesse“ in Schleifen ausgeführt werden.

Da alle genannten Diagrammformen auch die ggf. eingeklappten Prozesse enthalten, bietet die BPMN 2.0 somit nicht nur eine Wiedergabe der organisatorischen Abläufe oder implementierungsnahe Visualisierung, da auch Fehlerfälle behandelt werden können, sondern kann auch die Nachrichtenflüsse bzw. den Nachrichtenaustausch beschreiben. Zudem besteht die Möglichkeit, diese durch Annotationen (vgl. [All09, S.24, S.154]) zu nutzen, um beispielsweise die in Abschnitt 2.6 (Seite 38) beschriebene Notation dort zu hinterlegen.

4.1.2 Ausblick

Diagramme sind in BPMN 2.0 speziell dafür gedacht, in automatisch ausführbare Prozesse transformiert bzw. als solche verwendet zu werden. (vgl. [All09, S.13]) Dies könnte in einer späteren Arbeit als Erweiterung bis hin zu einer automatischen Konzeptverifikation oder zumindest einer automatischen Verifikation einzelner Teile führen. Ein ähnlicher Ansatz, bei dem Geschäftsprozesse und die zu erfüllenden Schutzziele bzw. -bedarfe erst in einer um Sicherheitsbedingungen erweiterten BPMN 2.0 modelliert und später durch mehrere XSL/XML Transformationen in Implementierungen für Webservices überführt werden, wird von Wolter et. al beschrieben.[Wol+09] Analog dazu sollte es möglich sein, einerseits die Abfolge bzw. Zusammenhänge eines Konzeptes oder Sicherheitsprotokolls in der BPMN 2.0 auszudrücken und andererseits die zu erfüllenden Schutzbedingungen und -ziele (die Anforderungen) ggf. mit einer Erweiterung oder direkt mithilfe der BPMN 2.0 zu modellieren, um beides später automatisch durch eine Transformation in ein Eingabeformat für Verifikationstools umzuwandeln. Damit wäre eine formale oder semi-formale Verifikation erheblich einfacher möglich, da ein Konzept nur auf Basis der relativ leicht verständlichen Diagramme modelliert werden müsste, um anschließend automatisch einer Verifikation unterzogen zu werden. Zudem könnten einmal verifizierte Teile ggf. als Bausteine weiter genutzt werden, um so Stück für Stück sicherere Gesamtkonzepte zu erarbeiten.

4.2 Anwendungsbeispiele

In den folgenden Abschnitten werden einige Auszüge aus Richters Konzept in der Notation bzw. Teilen davon dargestellt und erklärt. Die dargestellten Beispiele sind dabei i. d. R. bewusst unvollständig gehalten, da hier nur ein Gefühl für die Möglichkeiten der Notation gegeben werden soll. Dabei wird speziell auf die Verwendung der jeweils verwendeten Komponenten und ihre Funktion hingewiesen.

4.2.1 Überblick

Um über ein Konzept oder ein Netz- oder Sicherheitsprotokoll einen Überblick zu gewinnen, benötigt der Betrachter zumeist eine sehr abstrakte Darstellung der Kommunikationsbeziehungen. In Abbildung 4.1 ist dafür eine Darstellung mithilfe von eingeklappten "pools" gewählt, welche beispielhaft die abstrahierte Kommunikation des TMS mit der TA aufzeigt. Diese Form der Darstellung mit den Nachrichtenflüssen zwischen "pools" entspricht dabei den in Abschnitt 4.1.1 (Seite 67) genannten Kolloborationsdiagrammen. Es ist leicht vorstellbar, dass eine solche Form auch bei mehreren Kommunikationspartnern bzw. Rollen verwendbar ist, um z.B. die Nachrichtenflüsse der in Richter beteiligten und in Abschnitt 3.1 (Seite 44) beschriebenen Rollen (vgl. Abbildung 3.1 (Seite 46)) darzulegen.

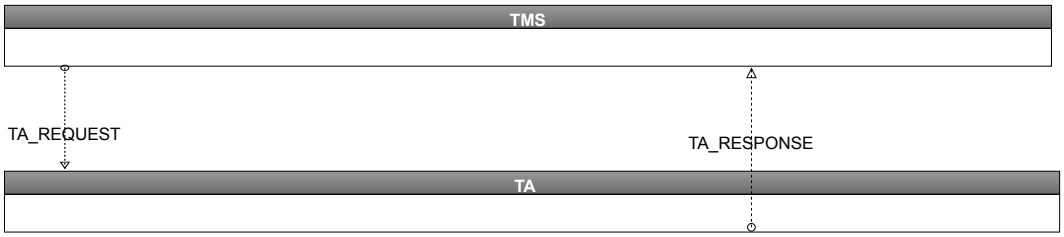

Abbildung 4.1 Beispielhafte TMS - TA Interaktion mit eingeklappten "pools"

Ferner wird in Abbildung 4.2 (Seite 70) in einem nicht eingeklappten "pool" des Betreibers die Verwendung der in Abschnitt 4.1.1 (Seite 67) genannten "swimlanes" gezeigt, um beispielhaft die aus Richters Konzept bekannte AEU und das TMS logisch der übergeordneten Rolle des Betreibers zuzuordnen. Dabei wird weiterhin der in Abbildung 4.1 gezeigte Nachrichtenfluss zwischen dem TMS und der TA visualisiert.

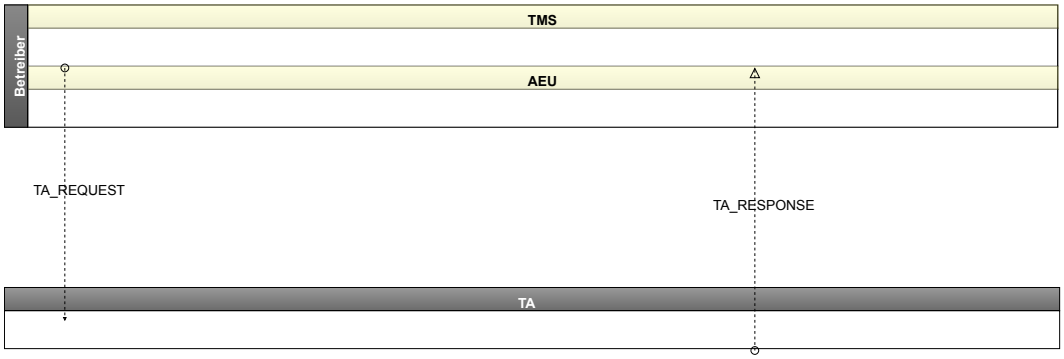

Abbildung 4.2 Beispielhafte Betreiber (TMS) - TA Interaktion

Auf Basis der Abbildung 4.2 folgt im nächsten Abschnitt 4.2.2 eine weitere Erhöhung des Detailgrades und somit eine weniger abstrahierte Ansicht der Zusammenhänge.

4.2.2 Detaillierte Prozessdarstellung

Durch die in Abschnitt 4.1.1 (Seite 67) und Abschnitt 4.2.1 (Seite 69) genannte Möglichkeit der "pools" bzw. "swimlanes", eingeklappt oder ausgeklappt darstellbar zu sein, wird in Abbildung 4.3 (Seite 71) selektiv sowohl der "pool" der TA als auch die "swimlane" des TMS aufgeklappt. Darunter verbirgt sich ein exemplarischer TMS-Betriebsprozess, bei dem nach dem Einschalten des TMS die in Abschnitt 3.1.4.3 (Seite 57) beschriebenen Initialisierungsaufgaben durchgeführt werden und später in den normalen Betrieb übergegangen wird.

Der in Abschnitt 3.1.4.3 (Seite 57) beschriebene sichere Startvorgang (vgl. auch Abschnitt 2.4.6.1 (Seite 29)) wird hier, um eine vereinfachte Darstellung zu erreichen, als eingeklappter „Unter-Prozess" dargestellt. Ebenfalls als „Unter-Prozess" ist der Prozess zur Zeitsynchronisation mit der TA enthalten. Dieser enthält jedoch die Besonderheit, dass die eingehenden und ausgehenden Nachrichtenflüsse von bzw. zur TA gekennzeichnet sind. Des Weiteren ist der Prozess zur Erfassung von Verkehrsverstößen (*MR erfassen*) als „Unter-Prozess" enthalten. Sowohl dieser als auch der Synchronisationsprozess verfügen dabei über die Besonderheit, dass sie gemäß ihrer internen Modellierung wiederholt werden, was durch den kleinen halbrunden Pfeil ausgedrückt wird.

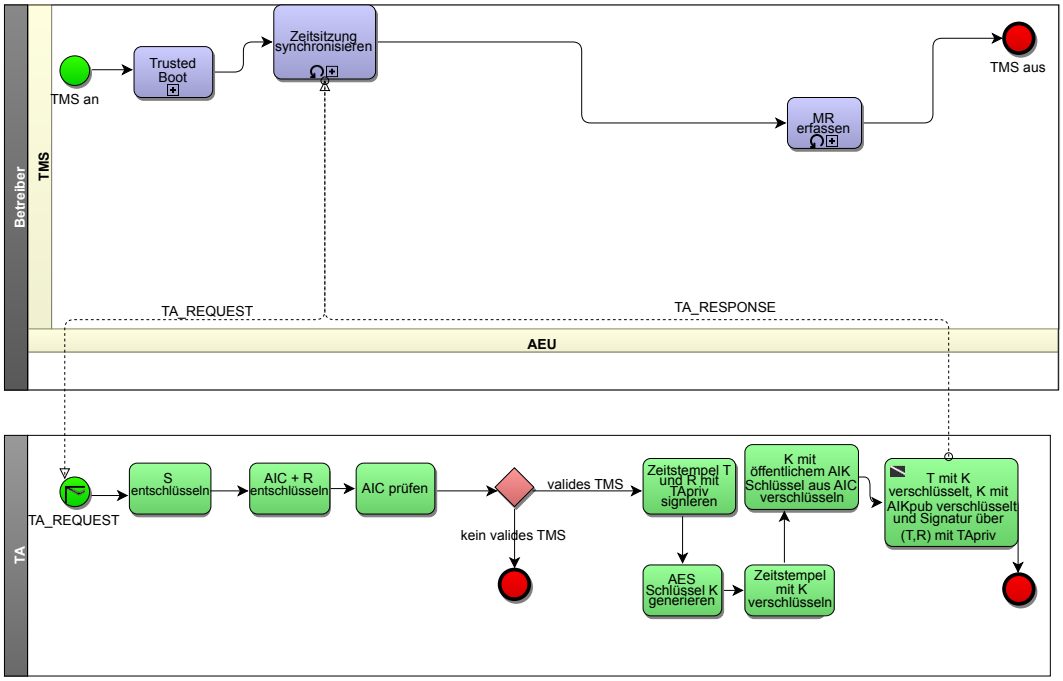

Abbildung 4.3 Beispielhafte Betreiber (TMS) - TA Interaktion

Die in Abbildung 4.3 eingeklappten „Unter-Prozesse“ *Zeitsitzung synchronisieren* und *MR erfassen* werden in Abbildung 4.4 (Seite 72) ebenfalls selektiv ausgeklappt, um einen noch höheren Detailgrad zu erreichen. Der „Unter-Prozess“ *Trusted Boot* wird dabei bewusst nicht ausgeklappt, da dieser nicht näher modelliert ist. Die BPMN 2.0 bietet durch nicht ausklappbare “pools”, “swimlanes” und „Unter-Prozesse“ die Möglichkeit, auch unbekannte Prozesse, die Teil eines größeren Ablaufes sind, aber deren konkreter Ablauf unbekannt ist, darzustellen.

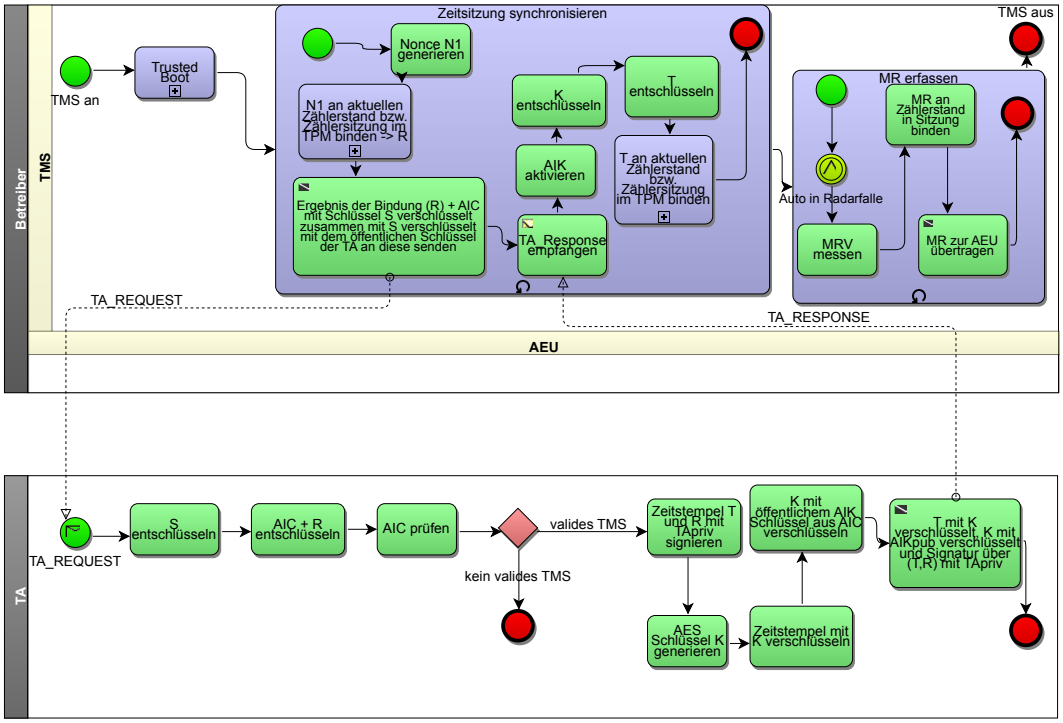

Abbildung 4.4 Beispielhafte Betreiber (TMS) - TA Interaktion

Letztlich kann durch das Auf- und Zuklappen die Darstellung eines Konzeptes oder Protokolls so konfiguriert werden, wie der Leser es benötigt. Um ein ungefähres Gefühl zu vermitteln, wie das Konzept von Richter in der hier vorgestellten Notation aussehen könnte, befindet sich im Anhang in Abbildung B.1 (Seite 96) eine Modellierung des gesamten Konzeptes. Die hier vorgeschlagenen Erweiterungen, um die zu erfüllenden Schutzbedingungen und -ziele als Anforderungen zu modellieren, sind hierbei nicht berücksichtigt, da die Abbildung lediglich eine grobe Orientierung bieten soll.

Kapitel 5

Fazit

Fazit

In der vorliegenden Arbeit wurde das von Richter [Ric09] beschriebene Konzept, zur automatisierten und rechtssicheren Erfassung von realen Ereignissen in digitaler Form, analysiert und auf seine Verständlichkeit hin untersucht. Dabei stellte sich im Laufe der Arbeit zunächst heraus, dass einige Punkte nicht genau genug bzw. widersprüchlich beschrieben sind und so das ursprüngliche Ziel, eine modulare und architekturell hochwertige Implementierung zu erzeugen (vgl. Abschnitt 1.3 (Seite 5)), nicht verfolgt werden konnte.

Stattdessen wurde zunächst eine Vereinfachung und Analyse des Konzeptes von Richter vorgenommen (vgl. Kapitel 3 (Seite 44)). Im Rahmen der Beschreibung bzw. Analyse des Konzeptes konnten einige Fehler oder Unklarheiten in der zu Grunde liegenden Arbeit von Richter und weiteren Arbeiten beispielhaft aufgezeigt werden. (vgl. Abschnitt 3.2 (Seite 59)) Es wurde deutlich, dass die Masse an zu überprüfenden Teilen im zu Grunde leigenden Konzept zu groß ist, um Kernbestandteile sinnvoll zu formalisieren. Im Rahmen der Dokumentenanalyse einiger TCG-Spezifikationen wurden sogar in einer Spezifikation sich widersprechende Aussagen gefunden. Auf der einen Seite wird der EK zum Verschlüsseln genutzt: “The SRK is generated by the TPM and the SRK pass phrase is encrypted using the EK [...] [TCG07, S.17]”. Auf der nächsten Seite wird ausgeschlossen, dass der EK jemals zur Verschlüsselung genutzt wird: “It is never used for encryption or signing. [TCG07, S.18]”.

Zudem entstand im Verlauf der Analyse des Konzeptes die Erkenntnis, dass eine reine Formalisierung der Kernbestandteile die Verständnisprobleme nicht grundsätzlich lösen würde. Ein Kernproblem zeigt sich dabei in der Komplexität der *Trusted Computing*-Konzepte und in der Darstellungsmächtigkeit der gewählten Abbildungen bzw. Beschreibungen, da diese nur jeweils einen ganz bestimmten Detailgrad visualisieren können. Beispielsweise zeigen manche Abbildungen von Richter (vgl. Abbildung 3.1 (Seite 46)) sehr abstrakt die beteiligten Entitäten bzw. Rollen, nicht jedoch deren Interaktion miteinander. Andererseits zeigt Abbildung 3.2 (Seite 62) eine konkrete Interaktion zwei-

er Entitäten mit einer vermeintlich genauen Angabe der Nachrichteninhalte, ist jedoch nicht konsistent zu Beschreibungen in anderen Darstellungsformen und enthält Fehler.

Aus den genannten Gründen heraus wurde als Erleichterung eine skalierbare Darstellungslösung auf Basis der BPMN 2.0 exemplarisch vorgestellt. Nach Meinung des Autors stellt diese speziell wegen der guten Skalierbarkeit hinsichtlich der Detailtiefe und bei gleichzeitiger Beibehaltung der inneren Konsistenz eine gute Alternative zur textuellen und auf verschiedenen, aber getrennten Wegen erstellten Beschreibungen eines Konzeptes dar. Auf Basis dieser lässt sich zudem, eine entsprechende Modellierung vorausgesetzt, eine Formalisierung und Verifikation erleichtern und ggf. automatisiert durchführen. Auch kann durch die Mächtigkeit der BPMN 2.0 in Verbindung mit den in Abschnitt 4.1 (Seite 66) beschriebenen *Process-Engines* ein Konzept ggf. simuliert werden.

Die vorgestellte BPMN 2.0 kommt zwar ursprünglich aus einer anderen Disziplin und wird für sicherheitsrelevante Modellierung bislang noch selten eingesetzt, es ist jedoch die Meinung des Autors, dass dies sich zukünftig ggf. mit einer speziell erweiterten BPMN 2.0-Variante ändern sollte. Denkbar wäre beispielsweise die Konzepte und Zusammenhänge der TCG langfristig in eine solche Form zu überführen. Dies hätte neben den Vorteilen durch die Formalisier-, Verifizier- und Simulierbarkeit auch den Vorteil, dass eine neutrale Instanz, beispielsweise ein Richter (die juristische Person ist hier gemeint), in die Lage versetzt werden würde, den Detailgrad der ihm präsentierten Ergebnisse Stück für Stück anzupassen, um so eine möglichst fundierte Entscheidung treffen zu können. Im Rahmen der Recherchen zu der vorliegenden Arbeit wurde das hierzu passende Zitat von Koenig gefunden :

> Es scheint vielmehr so, als hätten es ihre Entwickler darauf angelegt, die Technik selbst vor dem intellektuellem Zugriff unruhestiftender Juristen weitgehend abzusichern. Die Anwesenden Kryptologen werden dieses Konzept unter dem Stichwort „Security by Obscurity“ kennen. [Koe03, S.2]

Wird diese Aussage zusammen mit dem gefundenen Widerspruch und der aktuellen Aufregung (um “BEAST”, vgl. [Riz11]) um die Sicherheit des Transfer Layer Security (TLS)-Protokolls, das Protokoll zur Absicherung im Web, sowie der Tatsache, dass selbst seit Jahren spezifizierte sicherheitsrelevante Protokolle (z.B. TLS 1.2 [IET08] bzw. TLS 1.1

[IET06]) in aktuellen Programmen nicht implementiert sind (vgl. [Per09] bzw. [Bol11]) betrachtet, entsteht die Erkenntnis, dass sicherheitsrelevante Abläufe und Konzepte den Menschen einfacher zugänglich gemacht werden sollten.

Letztlich gilt es daher, die Verständlichkeit der *Trusted Computing*-Konzepte bzw. Spezifikationen durch den Einsatz der in dieser Arbeit vorgeschlagenen Darstellungsform (oder einer Erweiterung) zu erhöhen, um diese mehr Menschen zugänglich zu machen.

Anhang A

Verzeichnisse

A.1 Abbildungsverzeichnis

A.2 Literatur

[act11] activity.org. *Activiti*. online. BPMN2 Process Engine. 2011. URL: *http://www.activiti.org/* (abgerufen am 11. 10. 2011) (siehe S. 66).

[Allo9] Thomas Allweyer. *BPMN 2.0 Business Process Model and Notation*. Einführung in den Standard für die Geschäftsprozessmodellierung. Books on Demand GmbH, Norderstedt, 2009 (siehe S. 67, 68).

[AN95a] Martín Abadi und Roger Needham. *Prudent Engineering Practice for Cryptographic Protocols*. 1995. URL: *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.43.5306&rep=rep1&type=pdf* (abgerufen am 06. 10. 2011) (siehe S. 38).

[AN95b] Ross J. Anderson und Roger M. Needham. »Programming Satan's Computer«. In: *Computer Science Today*. 1995, S. 426–440. URL: *http://www.cl.cam.ac.uk/~rja14/Papers/satan.pdf* (abgerufen am 06. 10. 2011) (siehe S. 38).

[Ando3] Ross Anderson. *'Trusted Computing' Frequently Asked Questions*. online. TC / TCG / LaGrande / NGSCB / Longhorn / Palladium / TCPA. Aug. 2003. URL: *http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html* (abgerufen am 06. 10. 2011) (siehe S. 14).

[Ando4] Ross Anderson. »Cryptography and Competition Policy - Issues with 'Trusted Computing'«. In: *Economics of Information Security* 12 (2004). DOI: *10.1007/b116816*. URL: *http://www.cl.cam.ac.uk/~rja14/Papers/tcpa.pdf* (abgerufen am 06. 10. 2011) (siehe S. 14).

[Ando8] Ross Anderson. *Security Engineering*. John Wiley & Sons, 2008. URL: *http://www.cl.cam.ac.uk/~rja14/book.html* (abgerufen am 06. 10. 2011) (siehe S. 6, 37, 38).

[BCC04] Ernie Bricknell, Jan Camenisch und Liqun Chen. »Direct Anonymous Attestation«. In: (2004). URL: *http://eprint.iacr.org/2004/205.pdf* (abgerufen am 06. 10. 2011) (siehe S. 36).

[BHJP09] A. Brophy Haney, T. Jamasb und M.G. Pollitt. *Smart Metering and Electricity Demand: Technology, Economics and International Experience*. Cambridge Working Papers in Economics 0905. Faculty of Economics, University of

Cambridge, Feb. 2009. URL: *http://ideas.repec.org/p/cam/camdae/0905.html* (abgerufen am 06. 10. 2011) (siehe S. 3).

[Ble11] Blenta. *Yaoqiang BPMN Editor*. online. 2011. URL: *http://sourceforge.net/projects/bpmn/* (abgerufen am 11. 10. 2011) (siehe S. 66).

[BMJ01] Bundesjustizministerium (BMJ). *Strafprozeßordnung (StPO)*. online. Juni 2001. URL: *http://www.gesetze-im-internet.de/bundesrecht/stpo/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 41).

[BMJ09a] Bundesjustizministerium (BMJ). *Bundesdatenschutzgesetz (BDSG)*. online. Aug. 2009. URL: *http://www.gesetze-im-internet.de/bundesrecht/bdsg_1990/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40).

[BMJ09b] Bundesjustizministerium (BMJ). *Gesetz über Ordnungswidrigkeiten (OWiG)*. online. Juli 2009. URL: *http://www.gesetze-im-internet.de/bundesrecht/owig_1968/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40).

[BMJ09c] Bundesjustizministerium (BMJ). *Gesetz über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz - SigG)*. online. Juli 2009. URL: *http://www.gesetze-im-internet.de/bundesrecht/sigg_2001/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40).

[BMJ10a] Bundesjustizministerium (BMJ). *Grundgesetz für die Bundesrepublik Deutschland*. online. Juli 2010. URL: *http://www.gesetze-im-internet.de/bundesrecht/gg/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40).

[BMJ10b] Bundesjustizministerium (BMJ). *Straßenverkehrs-Ordnung (StVO)*. online. Dez. 2010. URL: *http://www.gesetze-im-internet.de/bundesrecht/stvo/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40, 41).

[BMJ11a] Bundesjustizministerium (BMJ). 2011. URL: *http://bundesrecht.juris.de/bundesrecht/ao_1977/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 2).

[BMJ11b] Bundesjustizministerium (BMJ). *Straßenverkehrsgesetz*. online. Juli 2011. URL: *http://www.gesetze-im-internet.de/bundesrecht/stvg/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 41).

[BMJ11c] Bundesjustizministerium (BMJ). *Zivilprozessordnung (ZPO)*. 2011. URL: *http://www.gesetze-im-internet.de/bundesrecht/zpo/gesamt.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40).

[Bol11] Nelson Bolyard. *Bug 565047 - (RFC4346) Implement TLS 1.1 (RFC 4346)*. online. Mozilla Bugtracker. Mai 2011. URL: *https://bugzilla.mozilla.org/show_bug.cgi?id=565047* (abgerufen am 13. 10. 2011) (siehe S. 76).

[bon11] bonitasoft.com. *Bonita Open Solution*. online. 2011. URL: *http://www.bonitasoft.com/products/BPM_downloads* (abgerufen am 11. 10. 2011) (siehe S. 66).

[BPM11] BPM Offensive Berlin (BPMOB). *BPMN 2.0 Poster*. online. 2011. URL: *http://www.bpmb.de/images/BPMN2_0_Poster_DE.pdf* (abgerufen am 11. 10. 2011) (siehe S. 94).

[BPS10] Carsten Bormann, Niels Pollem und Karsten Sohr. *Informationssicherheit 1, WS 2010/2011, Vorlesungsfolien*. 2010 (siehe S. 6, 8, 13).

[BSI09] BSI. *IT-Grundschutz-Kataloge*. online. Bundesamt für Sicherheit in der Informationstechnik. 2009. URL: *https://www.bsi.bund.de/ContentBSI/grundschutz/kataloge/kataloge.html* (abgerufen am 06. 10. 2011) (siehe S. 49).

[BVe83] Bundesverfassungsgericht (BVerfGE). *Volkszählung*. online. 1983. URL: *http://www.servat.unibe.ch/dfr/bv065001.html* (abgerufen am 06. 10. 2011) (siehe S. 40).

[Cam04a] Jan Camenisch. »Better Privacy for Trusted Computing Platforms«. In: (2004). IBM Research, Zurich Research Laboratory, CH-8803 Rüschlikon, Switzerland. URL: *http://www.zurich.ibm.com/~jca/papers/camenio4.pdf* (abgerufen am 06. 10. 2011) (siehe S. 36).

[Cam04b] Jan Camenisch. *Direct Anonymous Attestation: Achieving Privacy in Remote Authentication*. online. IBM Zurich Research Laboratory. Juli 2004. URL: *http://www.zurich.ibm.com/security/daa/daa-slides-ZISC.pdf* (abgerufen am 06. 10. 2011) (siehe S. 35, 36).

[Cha+07] David Challener, Kent Yoder, Ryan Catherman et al. *A practical guide to trusted computing*. First. IBM Press, 2007 (siehe S. 15, 16, 26–29, 34, 57).

[Cor11] Intel Corporation. *Intel® vPro™ Technology*. online. 2011. URL: *http://www.intel.com/content/www/us/en/architecture-and-technology/vpro/vpro-technology-general.html* (abgerufen am 06. 10. 2011) (siehe S. 24).

[Ecko9] Claudia Eckert. *IT-Sicherheit - Konzepte, Verfahren, Protokolle (6. Aufl.)* Oldenbourg, 2009, S. I–XIV, 1–981 (siehe S. 2, 6, 14, 16, 34, 36).

[Fero7] Jens Ferner. *Urteile des BVerfG zum Datenschutz*. online. 2007. URL: *http://www.datenschutzbeauftragter-online.de/das-bundesdatenschutzgesetz-bdsg/urteile-des-bverfg-zur-informationellen-selbstbestimmung/* (abgerufen am 06. 10. 2011) (siehe S. 40).

[Fico7] Barbara Fichtinger. »Trusted Infrastructures for Identities«. Magisterarb. Fachhochschule Hagenberg, Mai 2007. URL: *http://sit.sit.fraunhofer.de/smv/publications/download/Fichtinger_Trusted_Infrastructures_for_Identities.pdf* (abgerufen am 10. 10. 2011) (siehe S. 63).

[Fino6] Hal Finney. *testsuite-tcg-highlevel-tpm-Tspi_TPM_CreateIdentity.c and Privacy CA*. online. TrouSerS Mailinglist. Aug. 2006. URL: *http://permalink.gmane.org/gmane.comp.encryption.trousers.user/617* (abgerufen am 06. 10. 2011) (siehe S. 63).

[Fis+02] Stephanie Fischer-Dieskau, Rotraud Gitter, Sandra Paul und Roland Steidle. »Elektronisch signierte Dokumente als Beweismittel«. In: *Multimedia und Recht Zeitschrift für Informations-, Telekommunikations- und Medienrecht* 11 (2002), S. 709–713. URL: *http://www.uni-kassel.de/fb7/oeff_recht/publikationen/pubOrdner/Beweissicherheit_elektronischer_Dokumente.pdf* (abgerufen am 06. 10. 2011) (siehe S. 2).

[Gra07] David Grawrock. »Hello World«. In: (Sep. 2007). Intel vPro Expert Center Blog. URL: *http://communities.intel.com/community/openportit/vproexpert/blog/2007/09/25/hello-world* (abgerufen am 06. 10. 2011) (siehe S. 24).

[Hofo6] Mathis Hoffmann. »Der Beweiswert elektronischer Dokumente«. In: *DSWR - Zeitschrift für Praxisorganisation, Betriebswirtschaft und elektronische Datenverarbeitung* 03 (März 2006). S.60 ff, S.60 ff. URL: *http://www.jenanwalt.de/cms/files/der_beweiswert_elektronischer_dokumente.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40).

[IBM11] IBM. *Integrity Measurement Architecture (IMA)*. online. 2011. URL: *http://linux-ima.sourceforge.net/* (abgerufen am 06. 10. 2011) (siehe S. 31).

[IET06] IETF Network Working Group (IETF-NWG). *The Transport Layer Security (TLS) Protocol Version 1.1*. online. Apr. 2006. URL: *http://www.ietf.org/rfc/rfc4346.txt* (abgerufen am 13. 10. 2011) (siehe S. 76).

[IET08] IETF Network Working Group (IETF-NWG). *The Transport Layer Security (TLS) Protocol Version 1.2*. online. Aug. 2008. URL: *http://www.ietf.org/rfc/rfc5246.txt* (abgerufen am 13. 10. 2011) (siehe S. 75).

[ime11] imeikas. *imeikas / BPMN2-Editor-for-Eclipse*. online. 2011. URL: *https://github.com/imeikas/BPMN2-Editor-for-Eclipse* (abgerufen am 11. 10. 2011) (siehe S. 66).

[JLY04] Li Jiang, Da-You Liu und Bo Yang. »Smart home research«. In: *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*. Bd. 2. Aug. 2004, 659 –663 vol.2. DOI: *10.1109/ICMLC.2004.1382266*. (Abgerufen am 06. 10. 2011) (siehe S. 3).

[Koe03] Christian Koenig. *TCG und NGSCB auf dem Prüfstand des Wettbewerbsrechts*. online. Juli 2003. URL: *http://www.tkrecht.de/vortraege/bmwa2003/tc-vortrag-rede20030703.pdf* (abgerufen am 12. 10. 2011) (siehe S. 75).

[KR11] Nicolai Kuntze und Carsten Rudolph. »Secure digital chains of evidence«. 2011 (siehe S. 44).

[Kru09] Carsten Krumm. *SENSATION! BVerfG: Geschwindigkeitsmessungen, Abstandsmessungen etc. mit Video und Film (und auch Foto?) sind verfassungswidrig*. online. Richter am Amtsgericht. Aug. 2009. URL: *http://blog.beck.de/2009/08/20/senstation-bverfg-geschwindigkeitsmessungen-abstandsmessungen-mit-video-film-und-foto-sind-verfassungswidrig* (abgerufen am 06. 10. 2011) (siehe S. 40).

[LKS09] Andreas Leichner, Nicolai Kuntze und Andreas U. Schmidt. »Implementation of a Trusted Ticket System«. In: *Emerging Challenges for Security, Privacy and Trust*. 24th Ifip Tc 11 International Information Security Conference, SEC 2009, Pafos, Cyprus, May 18-20, 2009. Springer Berlin / Heidelberg, 2009. URL: *http://andreas.schmidt.novalyst.de/docs/TC_based_ticket_systems_IFIP_SEC_09_.pdf* (abgerufen am 06. 10. 2011) (siehe S. 63).

[Mlo09] Peter Mlodoch. »Neue Blitzer gegen Temposünder«. In: (Jan. 2009). Kölner Stadt-Anzeiger. URL: *http://www.ksta.de/html/artikel/1231945330054.shtml* (abgerufen am 06. 10. 2011) (siehe S. 47).

[Mül08] Thomas Müller. *Trusted Computing Systeme Konzepte und Anforderungen*. Springer Berlin / Heidelberg, 2008 (siehe S. 16–22, 24, 30–32, 45, 56).

[Per09] Jean-Yves Perrier. *Bug 480514 - Implement TLS 1.2 (RFC 5246)*. online. Mozilla Bugtracker. Feb. 2009. URL: *https://bugzilla.mozilla.org/show_bug.cgi?id=480514* (abgerufen am 13. 10. 2011) (siehe S. 76).

[PH10] Andreas Pfitzmann und Marit Hansen. *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management*. online. vo.34. Aug. 2010. URL: *http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_vo.34.pdf* (abgerufen am 13. 10. 2011) (siehe S. 9).

[PTB06] PTB. *Messgeräte im Straßenverkehr Geschwindigkeitsüberwachungsgeräte*. online. Nov. 2006. URL: *http://www.ptb.de/de/org/q/q3/q31/ptb-a/pa18-11.pdf* (abgerufen am 06. 10. 2011) (siehe S. 41).

[RC05] Jason Reid und William J. Caelli. »DRM, Trusted Computing and Operating System Architecture«. In: *ACSW Frontiers*. 2005, S. 127–136. URL: *http://www.crpit.com/confpapers/CRPITV44Reid.pdf* (abgerufen am 06. 10. 2011) (siehe S. 30).

[Ric09] Jennifer Richter. *Securing Digital Evidence*. Bachelorarbeit. 2009. URL: *http://sit.sit.fraunhofer.de/smv/publications/download/DigitalEvidenceThesis.pdf* (abgerufen am 06. 10. 2011) (siehe S. 3, 5, 6, 8, 10, 13, 15, 21, 36, 40, 44, 46–57, 59–63, 74, 96).

[Riz11] Juliano Rizzo. *BEAST: Surprising crypto attack against HTTPS*. online. Sep. 2011. URL: *http://www.ekoparty.org/2011/juliano-rizzo.php* (abgerufen am 13. 10. 2011) (siehe S. 75).

[RJP73] Roger R.Schell, Peter J.Downey und Gerald J. Popek. *Preliminary Notes On The Design Of Secure Military Computer Systems*. Approved for Public Release by Melvin B. Emmons. Electronic Systems Division Air Force System Command L.G. Hanscom Field Bedford, Massachusetts, Jan. 1973. URL:

http://www.ics.uci.edu/~djr/classes/ics280SSAT/readings/sche73.pdf (abgerufen am 06. 10. 2011) (siehe S. 15).

[RKR10] Jennifer Richter, Nicolai Kuntze und Carsten Rudolph. »Securing Digital Evidence«. In: *Fifth International Workshop on Systematic Approaches to Digital Forensic Engeneering*. 2010, S. 119–130. URL: *http://sit.sit.fraunhofer.de/smv/publications/download/EvidentialIntegrity.pdf* (abgerufen am 06. 10. 2011) (siehe S. 3, 44).

[Roß06] Alexander Roßnagel. *Rechtliche Aspekte der elektronischen Signatur*. online. provet - Projektgruppe verfassungsverträgliche Technikgestaltung, Universität Kassel. Mai 2006. URL: *http://www.lfd.m-v.de/dschutz/veransta/sigffm/rossnagel.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40).

[RSA82] Rivest, Shamir und Adleman. »A Method for Obtaining Digital Signatures and Public Key Cryptosystems«. In: *SIMMONS: Secure Communications and Asymmetric Cryptosystems*. 1982 (siehe S. 11, 20).

[Saf02] David Safford. »The Need for TCPA«. In: (Okt. 2002). IBM Research. URL: *http://www.research.ibm.com/gsal/tcpa/why_tcpa.pdf* (abgerufen am 06. 10. 2011) (siehe S. 14).

[Sch09] Hubert Schulze. *Die digitale Signatur und ihre juristische Bedeutung*. 2009. URL: *http://www.rrzn.uni-hannover.de/digi-signatur.html?&no_cache=1* (abgerufen am 06. 10. 2011) (siehe S. 2).

[Sir11] Sirrix. *TrustedGRUB*. online. developed by Sirrix AG. 2011. URL: *http://projects.sirrix.com/trac/trustedgrub* (abgerufen am 06. 10. 2011) (siehe S. 31).

[SPD05] Elaine Shi, Adrian Perrig und Leendert van Doorn. »BIND: A Fine-Grained Attestation Service for Secure Distributed Systems«. In: *IEEE Symposium on Security and Privacy*. 2005, S. 154–168. URL: *http://sparrow.ece.cmu.edu/group/pub/shi_perrig_vanDoorn.pdf* (abgerufen am 06. 10. 2011) (siehe S. 31).

[SSM10] Mario Strasser, Heiko Stamer und Jesus Molina. *Software-based TPM Emulator*. online. 2010. URL: *http://tpm-emulator.berlios.de/* (abgerufen am 06. 10. 2011) (siehe S. 28).

[Str04] Mario Strasser. »A Software-based TPM Emulator for Linux«. In: (2004). Department of Computer Science Swiss Federal Institute of Technology Zu-

rich Supervisors: Paul E. Sevinç Prof. Dr. David Basin. URL: *http://www.infsec.ethz.ch/people/psevinc/TPMEmulatorReport.pdf* (abgerufen am 06. 10. 2011) (siehe S. 28).

[TCG02] Trusted Computing Group (TCG). *Trusted Computing Platform Alliance (TCPA) Main Specification Version 1.1b*. online. Feb. 2002. URL: *http://www.trustedcomputinggroup.org/files/resource_files/64795356-1D09-3519-ADAB12F595B5FCDF/TCPA_Main_TCG_Architecture_v1_1b.pdf* (abgerufen am 06. 10. 2011) (siehe S. 27, 35).

[TCG05] Trusted Computing Group (TCG). *TCG PC Client Specific TPM Interface Specification (TIS)*. online. Version 1.20. Juli 2005. URL: *http://www.trustedcomputinggroup.org/files/resource_files/87BCE22B-1D09-3519-ADEBA772FBF02CBD/TCG_PCClientTPMSpecification_1-20_1-00_FINAL.pdf* (abgerufen am 06. 10. 2011) (siehe S. 27).

[TCG07] Trusted Computing Group (TCG). *TCG Specification Architecture Overview*. online. Aug. 2007. URL: *http://www.trustedcomputinggroup.org/files/resource_files/AC652DE1-1D09-3519-ADA026A0C05CFAC2/TCG_1_4_Architecture_Overview.pdf* (abgerufen am 06. 10. 2011) (siehe S. 15–17, 20–29, 32–35, 74).

[TCG11a] Trusted Computing Group Infrastructure Working Group (TCG-IWG). *A CMC Profile for AIK Certificate Enrollment*. online. März 2011. URL: *http://www.trustedcomputinggroup.org/files/resource_files/738DF0BB-1A4B-B294-D0AF6AF9CC023163/IWG_CMC_Profile_Cert_Enrollment_v1_r7.pdf* (abgerufen am 10. 10. 2011) (siehe S. 63).

[TCG11b] Trusted Computing Group (TCG). *About TCG*. online. 2011. URL: *http://www.trustedcomputinggroup.org/about_tcg* (abgerufen am 06. 10. 2011) (siehe S. 14, 16).

[TCG11c] Trusted Computing Group (TCG). *TCG PC Client Specific TPM Interface Specification (TIS)*. online. Apr. 2011. URL: *http://www.trustedcomputinggroup.org/files/static_page_files/C16ACEC4-1A4B-B294-D0076921424609F7/TCG_PCClientTPMSpecification_1-21_1-00_FINAL.pdf* (abgerufen am 06. 10. 2011) (siehe S. 27, 56).

[TCG11d] Trusted Computing Group (TCG). *TPM Main Part 1 Design Principles*. online. Specification Version 1.2. März 2011. URL: *http://www.trustedcomputinggroup.org/files/static_page_files/72C26AB5-1A4B-B294-D002BC0B8C062FF6/TPM%20Main-Part%201%20Design%20Principles_v1.2_rev116_01032011.pdf* (abgerufen am 06. 10. 2011) (siehe S. 28, 33, 34, 36, 57, 58, 60).

[TCG11e] Trusted Computing Group (TCG). *TPM Main Part 2 TPM Structures*. online. Specification Version 1.2. März 2011. URL: *http://www.trustedcomputinggroup.org/files/static_page_files/72C2B624-1A4B-B294-D0E07C5F7F49140D/TPM%20Main-Part%202%20TPM%20Structures_v1.2_rev116_01032011.pdf* (abgerufen am 06. 10. 2011) (siehe S. 28, 63).

[TCG11f] Trusted Computing Group (TCG). *TPM Main Part 3 Commands*. online. Specification Version 1.2. März 2011. URL: *http://www.trustedcomputinggroup.org/files/static_page_files/72C33D71-1A4B-B294-D02C7DF86630BE7C/TPM%20Main-Part%203%20Commands_v1.2_rev116_01032011.pdf* (abgerufen am 06. 10. 2011) (siehe S. 28).

[Tec05] Infineon Technologies. *Trusted Platform Module (TPM1.2 PC)*. online. Okt. 2005. URL: *http://www.infineon.com/cms/en/product/chip-card-and-security-ics/embedded-security/trusted-computing/trusted-platform-module-tpm1.2-pc/channel.html?channel=ff80808112ab681d0112ab6921ae011f* (abgerufen am 06. 10. 2011) (siehe S. 24).

[Tro11] TrouSerS. *TrouSerS - The open-source TCG Software Stack*. online. 2011. URL: *http://trousers.sourceforge.net/* (abgerufen am 06. 10. 2011) (siehe S. 63).

[Tüb04] Bürgermeisteramt Tübingen. *Mitteilung im: Verkehrsplanungs- und Umweltausschuss*. online. Geschwindigkeitsmessungen - Rechtliche Grundlagen, Gesch. Z.: 31/150-00. Apr. 2004. URL: *http://www.tuebingen.de/ratsdokumente/2004_32.pdf* (abgerufen am 06. 10. 2011) (siehe S. 40).

[Woh00] Petra Wohlmacher. »Sicherheitsanforderungen und Sicherheitsmechanismen bei IT-Systemen«. In: *EMISA Forum* 10.1 (2000). URL: *http://subs.emis.de/LNI/EMISA-Forum/Volume20_1/wohlmacher.pdf* (abgerufen am 06. 10. 2011) (siehe S. 2).

[Wol+09] Christian Wolter, Michael Menzel, Andreas Schaad et al. »Model-driven business process security requirement specification«. In: *Journal of Systems Architecture* 55.4 (2009). <ce:title>Secure Service-Oriented Architectures (Special Issue on Secure SOA)</ce:title>, S. 211 –223. ISSN: 1383-7621. DOI: *10.1016/j.sysarc.2008.10.002*. URL: *http://www.sciencedirect.com/science/article/pii/S1383762108001471* (abgerufen am 13. 10. 2011) (siehe S. 68).

[WS10] Robert Weihmann und Claus Peter Schuch. *Kriminalistik: Für Studium, Praxis, Führung*. Hilden: Verlag Deutsche Polizeiliteratur, 2010. URL: *http://www.weihmann.info/images/Kriminalistik/Kapitel%2013,%20Alibi.pdf* (abgerufen am 06. 10. 2011) (siehe S. 3).

A.3 Abkürzungen

AEU

Archiving and Evaluation Unit

S. 45, 46, 48, 49, 51, 54, 56, 57, 59, 69

AIC

Attestation Identity Credential

S. 24, 34, 35, 56–59, 61

AIK

Attestation Identity Key

S. 20, 24, 34–36, 56–61, 63

API

Application Programming Interface

S. 27

BIOS

Basic Input Output System

S. 15, 17, 30

CCRED

Conformance Credential

S. 23, 34, 35, 56

CRTM

Core Root of Trust for Measurement

S. 17, 30

DAA

Direct Anonymous Attestation

S. 36

DRM

Digital Rights Management

S. 14

EK

Endorsement Key

S. 18–25, 34–36, 54, 56, 60, 61, 74

EKCRED

Endorsement Credential

S. 20, 22, 23, 56

KCM

Key Cache Manager

S. 26

MR

Measurement Record

S. 46, 48, 49, 51, 54, 55, 57–59

MRV

Measurement Record Value

S. 47–50, 59

OAS

Owner Authorization Secret

S. 19, 21

PC

Personal Computer

S. 26

PCA

Privacy Certification Authority

S. 34–36, 46, 49, 57, 60, 61, 63

PCR

Platform Configuration Register

S. 20, 25, 26, 30, 33, 35, 52, 59

PCRED

Platform Credential

S. 20, 23, 24, 34, 35, 56

RAM

Random Access Memory

S. 16

RNG

Random Number Generator

S. 25, 33

RPC

Remote Procedure Call

S. 28

RTM

Root of Trust for Measurement

S. 17

RTR

Root of Trust for Reporting

S. 17

RTS

Root of Trust for Storage

S. 17, 18, 21

SML

System Management Log

S. 53

SRK

Storage Root Key

S. 18, 20, 22, 25

SSL

Secure Sockets Layer

S. 38

TA

Timing Authority

S. 46, 49, 57–59, 61, 69–72, 78

TBB

Trusted Building Block

S. 16, 17, 19, 23

TCG

Trusted Computing Group

S. 13, 14, 16, 18–29, 31–34, 36, 44, 52, 53, 56, 59, 60, 63, 74, 75

TCP

Trusted Computing Platform

S. 17, 18, 21–24, 29

TCS

TCG Core Services

S. 28, 29, 32

TDDL

TCG Device Driver Library

S. 28

TLS

Transfer Layer Security

S. 38, 75

TMS

Traffic Management System

S. 44–59, 61, 63, 69–72, 78

TP

Trusted Platform

S. 13, 15, 16, 26, 35

TPM

Trusted Platform Module

S. 5, 14–30, 32–36, 45, 46, 48, 53–61, 63

TSP

TCG Service Provider

S. 28, 29, 32

TSPI

TCG Service Provider Interface

S. 57

TSS

TCG Software Stack

S. 15, 19, 21, 27, 53, 54, 63

VCRED

Validation Credential

S. 23, 24, 56

A.4 Glossar

BPMN 2.0

Die Business Process Modelling Notation ist ein Standard zur Modellierung von Geschäftsprozessen auf Basis eines XML artigen Dateiformates. Eine gute Übersicht über die Notation bietet dieses BPMN2 Poster aus Berlin [BPM11].

S. 66–68, 71, 75

Nonce

Unter einer Nonce wird ein zufälliger Wert verstanden, der bspw. verhindern soll, dass eine auf dem Kommunikationskanal mitgeschnittene Nachricht im Rahmen eines Replay-Angriffes wieder abgespielt und verwendet werden kann, um dieselbe Aktion auszulösen, die die Nachricht ursprünglich ausgelöst hat.

S. 33, 34, 39, 57, 58, 61

Anhang B

Abbildungen

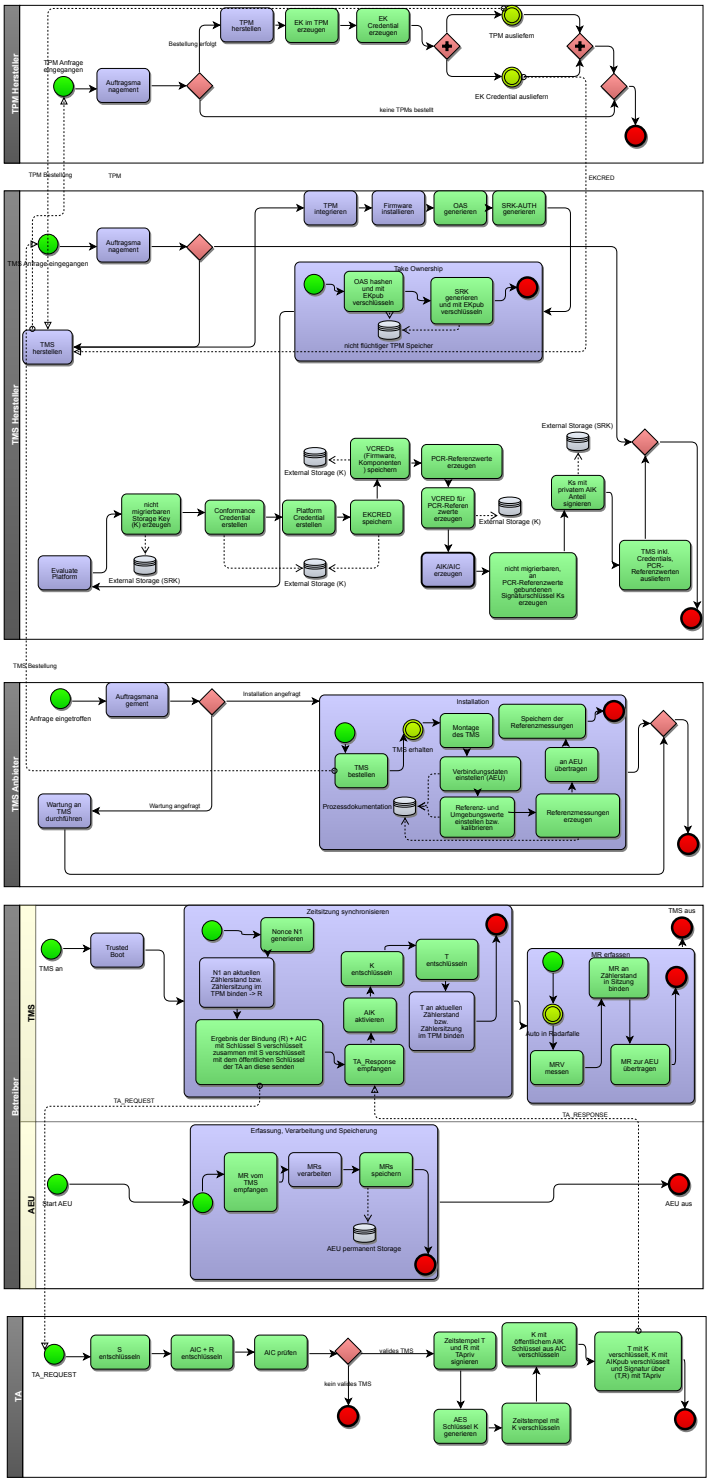

Abbildung B.1 Beispielhafte Darstellung des Konzeptes von Richter [Ric09]