fhwedel
UNIVERSITY OF APPLIED SCIENCES

Trusted Cloud Computing Platform

Seminararbeit

an der Fachhochschule Wedel

Vorgelegt von: **Wolf-Steffen Fröhlich**

Ma. Nr.: 9604

Gutachter: **Prof. Dr. Gerd Beuster**

Feldstraße 143 - 22880 Wedel

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Ein allgegenwärtiges Thema der Informationstechnik ist die IT-Sicherheit, geprägt von immer neuen und immer dramatischeren Meldungen über neue Schadsoftware und Datendiebstahl. 2011 wurden vom IT-Sicherheitsunternehmen Symantec rund 403 Millionen neue Schadsoftwarevarianten identifiziert.[1] Der 2011 bekannt gewordene Datendiebstahl bei Sony umfasst rund 77 Millionen Personendaten, wobei nicht ausgeschlossen werden kann, dass darunter auch Kreditkarteninformationen waren.[2] Typische Maßnahmen zur Absicherung der IT-Infrastruktur sind derzeit Firewalls, Virenscanner und Proxyserver. Die genannten Vorfälle zeigen jedoch, dass dadurch keine absolute Sicherheit erreicht werden kann. Gleichwohl eine absolute Sicherheit unmöglich ist, gibt es Ansätze, die diesem Problem auf gänzlich andere Weise entgegentreten. Einer dieser Ansätze ist das Trusted Computing, welches federführend von der Trusted Computing Group verfolgt und vorangetrieben wird.

Die grundsätzliche Idee lässt sich aus der Automatentheorie ableiten: Ein System befindet sich stets in einem Zustand von dem es durch Aktionen, wie zum Beispiel das Starten eines Programms, in einen Folgezustand wechselt. Wenn alle gültigen Zustände protokolliert sind und die Zustände auf ihre Gültigkeit hin überprüft werden können, ist es möglich Schadsoftware-Infektionen zuverlässig zu identifizieren und passende Maßnahmen einzuleiten.

Ein vollkommen anderer Trend ist das sogenannte Cloud Computing. Im Privatkundenbereich ist die Nutzung von Diensten wie Dropbox, iCloud oder Google Docs mittlerweile eine Selbstverständlichkeit. Im Geschäftskundenbereich erfreuen sich individualisierte Infrastructure-as-a-Service-Lösungen, wie zum Beispiel die Amazon WebServices, einer ebenso großen Beliebtheit. Infrastructure-as-a-Service-Lösungen stellen einem Nutzer durch Virtualisierungstechnik Rechenleistung anstatt Hardware bereit. Auf welchem physikalischen System dann eine Anwendung ausgeführt wird ist nicht ersichtlich. Die Nutzer profitieren von Kosteneinsparungen, welche die Anbieter durch Systemkonsolidierung erreichen. Im besten Fall wird die zur Verfügung stehende Rechenleistung optimal auf alle Nutzer verteilt – der Überschuss an nicht genutzter Rechenleistung wird minimiert.

[1] Vgl. Symantec – Internet Security Threat Report 2011, Volume 17.
[2] Vgl. Gieselmann – Angriff auf das Playstation Network: Worauf Kunden jetzt achten sollten. Online im Internet: URL: http://heise.de/-1233709 (Abgerufen am 9.2.2013)

Das größte Problem und damit Hemmnis des kommerziellen Cloud Computings ist der Datenschutz. Die von Unternehmen verarbeiteten Daten unterliegen bis auf wenige Ausnahmen entweder datenschutzrechtlichen Auflagen oder sind im Sinne des Geschäftsgeheimnisses schützenswert. Da keine Kontrolle über das physikalische System besteht, kann nicht sichergestellt werden, dass kein Zugriff durch Unbefugte stattfindet. Zudem ist Schadsoftware, welche direkt auf der physikalischen Maschine ausgeführt wird, eine zusätzliche Bedrohung für das Kundensystem.

In dieser Seminararbeit soll das Konzept der Trusted Cloud Computing Platform vorgestellt werden, welches den Ansatz des Trusted Computings auf Infrastructe-as-a-Service-Lösungen überträgt und damit eine vertrauenswürdige und geschützte Ausführungsumgebung für virtualisierte Systeme mit sensiblen Unternehmensdaten bereitstellt. Mit Hilfe der klassischen IT-Sicherheitsziele – Vertraulichkeit, Integrität und Verfügbarkeit – soll die Frage „Kann die Sicherheit von Cloud Anwendungen durch Trusted Computing erhöht werden?“ beantwortet werden.

2 Technologischer Hintergrund

2.1 Trusted Computing

Der Ursprung des Trusted Computing Ansatzes liegt in einem intern verwendeten Bewertungsstandard des US-Verteidigungsministeriums zur Zertifizierung von Computersystemen. Auszüge daraus sind heute noch in den *Common Criteria for Information Technology Security Evaluation* (CC) zu finden.[3] Die ersten Veröffentlichungen mit den wesentlichen Ideen des heutigen Trusted Computings erschienen in den neunziger Jahren von Tygar (1991)[4] und Arbaugh et al. (1997). Die von Arbaugh et al. modellierte Architektur namens AEGIS umfasste bereits eine Hardware-Erweiterung, welche als Vertrauensanker diente und mittels Hashwerten eine Vertrauenskette aufbaute.[5] Diesen Ansatz hat die *Trusted Computing Platform Alliance* (TCPA) aufgegriffen und fortgeführt, bis sie sich schließlich aufgrund von öffentlicher Kritik 2003 auflöste. Die Resultate der TCPA übernahm die 2003 gegründete und heute noch aktive *Trusted Computing Group* (TCG), welche federführend den Standard des Trusted Computings weiter entwickelt.[6]
Da die Trusted Computing Group der Hauptakteur in der Entwicklung des Standards ist, soll ihre Definition des Trusted Computings zugrunde gelegt werden:

> „Trust in the context of 'Trusted Platforms' is the expectation that a device will behave in a particular manner for a specific purpose."[7]

Diese abstrakte Auslegung des Begriffs spiegelt den Kerngedanken treffend wieder: Anwender können durch Trusted Computing sicherstellen, dass sich ihr System für einen bestimmten Zweck in einer bestimmten Art und Weise verhält. Dazu wird das ausgeführte System als eine Folge von Zuständen verstanden, welche durch die Ausführung von Programmen geändert werden. Der Zustand, in dem sich ein System zu einem Zeitpunkt befindet, kann auf seine Vertrauenswürdigkeit hin überprüft werden. Schlägt diese Prüfung fehl, ist das System nicht vertrauenswürdig und die Ausführung

[3] Vgl. Müller 2008 – S. 13 f.
[4] Siehe Tygar, Yee.
[5] Siehe Arbaugh et al.
[6] Vgl. Müller 2008 – S. 13.
[7] Vgl. ISO/IEC 11889-1:2009 – S. 4.

kritischer Anwendungen, wie zum Beispiel einer Onlinebanking-Software, wird verhindert.

2.1.1 Technische Realisierung

Die Aufgabe des Vertrauensankers übernimmt eine zusätzliche Hardwarekomponente auf dem Mainboard des Computers, das sogenannte *Trusted Platform Modul* (TPM). Das TPM arbeitet als autarker Computer im Computer des Anwenders und verfügt über alle notwendigen Komponenten und Schnittstellen, um eine Vertrauenskette aufzubauen.
Der Aufbau des TPM ist in Abb. 1 schematisch dargestellt:

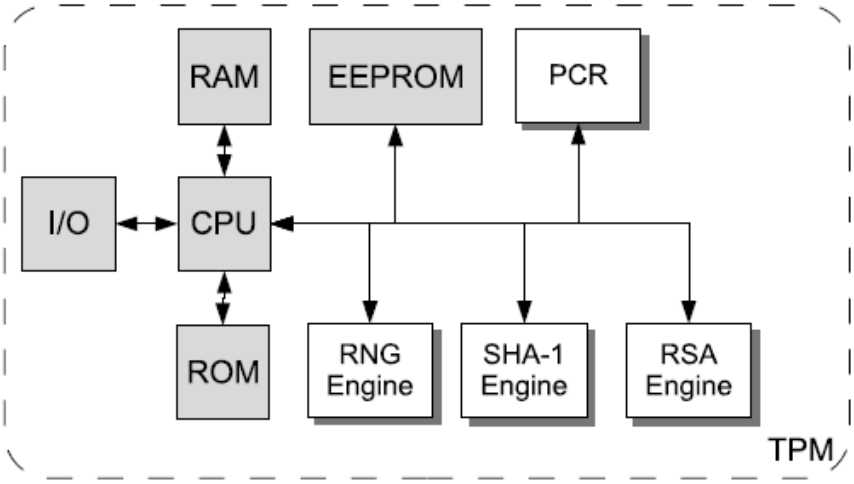


Abb. 1: Schematischer Aufbau des TPM[8]

Um unabhängig von anderen, gegebenenfalls manipulierten oder manipulierenden Komponenten zu sein, verfügt das TPM über eine eigene CPU als Prozessor, RAM als Arbeitsspeicher, ein EEPROM als nicht flüchtigen Speicher und Schnittstellen (in der Abbildung I/O für Input/Output), um mit dem Computer des Anwenders zu kommunizieren. In weiß dargestellt sind zusätzliche Komponenten, die für das Trusted Computing benötigt werden: Die RNG Engine erzeugt echte Zufallszahlen, die SHA-1 Engine berechnet Hashwerte nach dem SHA1-Algorithmus und die RSA Engine generiert Schlüsselpaare auf Basis des RSA-Verfahrens. Das *Platform Configuration Register* (PCR) ist ein spezieller, flüchtiger Speicher innerhalb des TPM. Es speichert mindestens 16 von der SHA-1 Engine erzeugte Hashwerte, welche den Systemzustand repräsentieren. Dazu wird vor der Ausführung jedes Programms stets dessen Hashwert

[8] Entnommen aus Müller 2008 – S. 31.

ermittelt. Dieser Hashwert wird nach der Formel 1 mit dem vorhandenen Hashwert im PCR verknüpft, aus dem Ergebnis ein neuer Hashwert gebildet und im PCR gespeichert. Dadurch bleiben die vorangegangenen Systemzustände erhalten und eine Folge von Zuständen wird aufgebaut.[9]

$$PCRi\ New\ =\ HASH(PCRi\ Old\ value\ ||\ value\ to\ add)$$

Formel 1: Berechnung des Hashwerts für das PCR[10]

Die vorhandenen Register des PCR sind für die folgenden Werte vorgesehen[11]:

- Register 0-4: BIOS, ROM, Memory Block Register
- Register 5-7: OS loaders
- Register 8-15: Operating System (OS)
- Register 16: Debug
- Register 17-22: Localities, Trusted OS
- Register 23: Applications specific

Im nichtflüchtigen Speicher werden insbesondere RSA-Schlüsselpaare gespeichert, die unterschiedliche Funktionen besitzen:
Der *Endorsement Key* (EK) ist ein 2048Bit langes RSA-Schlüsselpaar, welches vom Hersteller beim Fertigungsprozess des TPM erzeugt wird. Der private Schlüssel ist nicht veränder- oder auslesbar und kann zur eindeutigen Identifikation des TPM genutzt werden. Zusammen mit einem EK-Zertifikat vom Hersteller lässt sich das TPM als von einem vertrauenswürdigen Hersteller gefertigtes Modul identifizieren und kann dadurch selbst als vertrauenswürdig eingestuft werden. Allerdings bringt ein eindeutig identifizierbarer EK datenschutzrechtliche Probleme mit sich: Es lässt sich mit Hilfe der verschlüsselten Nachrichten ein Bewegungsprofil des Computers bzw. Nutzers erstellen, da diese nur mit dem zum privaten EK passenden öffentlichen Schlüssel entschlüsselt werden können.[12] Daher wird für die reguläre Kommunikationsverschlüsselung und –signatur ein zusätzlicher *Attestation Identity Key* (AIK) als Alias verwendet. Um den Datenschutz zu gewährleisten, darf der AIK nicht mit dem EK in Verbindung gebracht werden können. Dadurch entfällt allerdings die Vertrauenswür-

[9] Vgl. ISO/IEC 11889-2:2009 – S. 8 ff.
[10] Vgl. ISO/IEC 11889-2:2009 – S. 18.
[11] Vgl. Müller 2008 – S. 36.
[12] Vgl. Müller 2008 – S. 34.

digkeit, weshalb eine weitere Instanz – eine Trusted Third Party oder auch Privacy-Certificate Authority – hinzugezogen werden muss. Nur die Privacy Certificate Authority überzeugt sich anhand des EK-Zertifikats von der Vertrauenswürdigkeit eines Systems und erstellt daraufhin ein AIK-Zertifikat, welches die Vertrauenswürdigkeit gegenüber Kommunikationspartnern bestätigen soll.[13]

2.1.2 Der vertrauenswürdige Startprozess

Der Startprozess einer Trusted Platform beginnt nach dem Einschalten des Systems mit dem Core Root of Trust, einem Programm, welches noch vor dem BIOS gestartet wird. Das Core Root of Trust prüft den Zustand der Systemkomponenten, einschließlich des BIOS und übergibt die Ergebnisse dem TPM zur Speicherung. Anschließend wird das BIOS aufgerufen, welches die Systemkomponenten auf ihre Funktionsfähigkeit überprüft und den Hashwert des Bootloaders aus dem Master Boot Record des ausgewählten Startmediums an das TPM übermittelt. Danach erhält der Bootloader die volle Kontrolle über das System und muss sicherstellen, dass dieses Verfahren fortgesetzt wird. Bevor also ein zweiter, erweiterter Bootloader vom Speichermedium geladen wird, muss der erste Bootloader zunächst dessen Hashwert ermitteln und an das TPM übergeben. Dieses Verfahren wird bis zur Applikationsebene fortgesetzt, vorausgesetzt das Betriebssystem unterstützt diese Funktion. Auf diese Weise wird Stück für Stück eine sogenannte Vertrauenskette aufgebaut.[14] Die Vertrauenskette wird von Kommunikationspartnern zur Überprüfung der Vertrauenswürdigkeit des Systemzustands genutzt. Dazu nimmt der Kommunikationspartner Kontakt mit dem Attestation Client des Systems auf und fordert dessen PCR an. Der Attestation Client ist eine elementare Komponente eines vertrauenswürdigen Betriebssystems und ermöglicht die direkte (Netzwerk-)Kommunikation mit dem TPM. Zur Absicherung wird die gesamte Kommunikation mit den AIKs signiert und verschlüsselt. Mit der Privacy Certificate Authority lassen sich die AIKs und Zertifikate auf Gültigkeit überprüfen. Die Vertrauenswürdigkeit des Kommunikationspartners kann verifiziert werden, indem der Systemzustand beispielsweise mit einer Referenzdatenbank verglichen wird. Wenn die Datenbank nur gültige Systemzustände enthält, gilt das geprüfte System bei Übereinstimmung als vertrauenswürdig und die weitere Kommunikation kann gestattet werden.[15]

[13] Vgl. Müller 2008 – S. 35.
[14] Vgl. Müller 2008 – S. 54 f.
[15] Vgl. Müller 2008 – S. 74 f.

2.2 Virtualisierung

Mit der Virtualisierungstechnik wird die Hardware abstrahiert und als logische Ressource für Anwendungen oder ganze Betriebssysteme bereitgestellt. Eine treffende Definition lautet:

> „Virtualization is a way to abstract applications and their underlying components away from the hardware supporting them and present a logical or virtual view of these resources.“[16]

Die Virtualisierung ist keine neue Technik, sondern wurde bereits in den sechziger Jahren von IBM entwickelt und seitdem kontinuierlich verbessert. Die damaligen Beweggründe waren dieselben wie heute, so dass heute die Virtualisierungstechnik für IBM-Mainframes eine Selbstverständlichkeit ist.[17]
Nicht nur auf Mainframes wird Virtualisierungstechnik eingesetzt, sondern auch im Endkundenbereich, beispielsweise um ein Betriebssystem parallel auf einem System zu betreiben. Großes kommerzielles Interesse löste die Vollvirtualisierung von Windows- und Linux-Serversystemen auf konventioneller Hardware aus, da ein großes Einsparpotential erwartet wurde.
Eine Software, die *virtuelle Maschinen* (VM) ausführt, wird als Hypervisor bezeichnet. Populäre Produkte sind Microsoft Hyper-V und der VMWare Player. Eine Spezialform sind sogenannte Bare-Metal-Hypervisor. Ein Bare-Metal-Hypervisor ist ein eigenständiges Betriebssystem, dessen einzige Aufgabe die Ausführung von VMs ist.[18] Beispiele hierfür sind VMWare’s ESX(i) und der Citrix XenServer. Die Administration dieser Systeme erfolgt oftmals über eine Clientanwendung, die über das Netzwerk mit dem Bare-Metal-Hypervisor[19] kommuniziert. Direkt am System kann, wenn überhaupt, nur über eine Konsole gearbeitet werden. Da das Betriebssystem keine weiteren Aufgaben übernimmt, können die vorhandenen Ressourcen den VMs direkt zur Verfügung gestellt werden.
Neben dem eigentlichen Betrieb der VM ist die wesentliche Aufgabe des Hypervisors die Abschottung der VMs untereinander. Dazu werden zwei Verfahren eingesetzt: Zum

[16] Kusnetzky 2011 – S. 1.
[17] Travassos – Virtualization Trends Trace Their Origins Back to the Mainframe. Online im Internet: URL: http://ibmsystemsmag.com/mainframe/administrator/Virtualization/history_virtualization (Abgerufen am 9.2.2013).
[18] Göpel 2012 – S. 8.
[19] Im Folgendem mit Hypervisor abgekürzt.

einen die Emulation der abzuschottenden Komponenten – beispielsweise des Festplatten – oder USB-Controllers. Das Gastsystem verwendet dafür einen speziellen Treiber, welcher den Datenstrom passend an den Hypervisor zur weiteren Verarbeitung weiterleitet. Bei der Emulation von CPU oder RAM kommt es dann allerdings zu einem deutlichen Performanceverlust. Deswegen wird zum anderen ein direkter Zugriff auf die Hardware ermöglicht aber nur unter bestimmten Voraussetzungen. Damit eine VM beispielsweise nicht auf die Speicherbereiche einer anderern VM zugreifen kann, muss der Hypervisor die Zugriffe kontrollieren und überwachen können. Dazu wird das Konzept der Privilegierungsstufen erweitert. Konventionelle x86er-Prozessoren besaßen ursprünglich 4 Privilegierungsstufen (auch und im Folgenden CPU-Ringe genannt), mit jeweils unterschiedlicher Priorität. Der CPU-Ring 0 hatte die höchste Priorität und wurde vom Betriebssystem für Verwaltungsaufgaben (zum Beispiel Scheduler oder Speicherverwaltung) genutzt. Anwendungsprogramme hingegen wurden im CPU-Ring 3 mit deutlich niedrigerer Priorität ausgeführt. Dadurch hatte beispielsweise der Scheduler die Berechtigung laufende Anwendungsprogramme zu pausieren. Normale Anwendungsprozesse hingegen durften derartige Operationen nicht ausführen. Gängige Bezeichnungen für den CPU-Ring 0 und den CPU-Ring 3 sind auch Kernel-Mode und User-Mode.[20]

Ein Hypervisor bringt das Problem mit sich, dass eigentlich dieser die höchste Priorität haben sollte, um die VMs überwachen und steuern zu können. Wenn der Hypervisor im CPU-Ring 0 arbeiten würde, müsste das Betriebssystem in den CPU-Ring 1 verschoben werden. Da das Betriebssystem dann allerdings nicht mehr auf bestimmte Operationen zugreifen könnte, würden kritische Fehler auftreten. Damit das System dann nicht abstürzt, müssten diese Fehler umfangreich abgefangen und behandelt werden. Dies hätte wieder einen Performanceverlust zur Folge. Aus diesem Grund haben die CPU-Hersteller Intel und AMD[21] ihre CPUs um einen zusätzlichen Ring erweitert. Dieser CPU-Ring -1 hat nun die höchste Priorität und kann über Prozesse des CPU-Rings 0 verfügen, in dem die virtualisierten Betriebssysteme ausgeführt werden.[22]

[20] Göpel 2012 – S. 3 ff.
[21] Diese Erweiterung nennt Intel VT-x und AMD Secure Virtual Machine (SVM).
[22] Göpel 2012 – S. 4 ff.

2.3 Angriffsvektoren

2.3.1 Beispielszenario

Die Firma Beispiel GmbH verkauft Glühbirnen zu Discountpreisen. Dazu werden gezielt große Chargen von ausgewählten Lieferanten bezogen und per Versand weiterverkauft. Die Auswahl der aktuellen Lieferanten erfolgt über eine selbst entwickelte Software, die automatisch bei mehreren Anbietern die Preise ermittelt, vergleicht und das günstigste Angebot auswählt. Eine zentrale Datenbank verwaltet die Kundendaten und die jeweiligen Konditionen. Zur weiteren Kostenreduktion möchte die IT-Abteilung den Applikations- und Datenbankserver virtualisiert in ein Rechenzentrum auslagern.

Die Angriffsvektoren zielen auf die klassischen Grundwerte der Informationssicherheit[23] ab. Damit der Fokus auf den Spezifika des Dienstes Infrastructure-as-a-Service liegt, werden Angriffe, die bei einer herkömmlichen Infrastruktur denkbar sind, nicht betrachtet.

2.3.2 Datenvertraulichkeit

Die Vertraulichkeit von Daten ist sichergestellt, solange die Daten nur demjenigen zur Verfügung stehen, für den sie bestimmt sind.[24] Können Personen auf Daten zugreifen, die nicht für sie bestimmt sind, kann auf vielfältige Art und Weise Schaden entstehen. In dem oben beschriebenen Szenario wäre es beispielsweise denkbar, dass ein Mitarbeiter des Rechenzentrums eine Kopie der virtuellen Maschine anfertigt und diese an einen Konkurrenten des Unternehmens weiterreicht. Der entstandene Schaden ergibt sich aus den verlorenen Wettbewerbsvorteilen, da dem Konkurrenten nun sowohl die selbst entwickelte Software, als auch die Kundendaten der Beispiel GmbH zur Verfügung stehen.

2.3.3 Datenintegrität

Datenintegrität bezeichnet die Korrektheit von Daten.[25] So ist die Beispiel GmbH stets auf korrekte Daten in ihrer Kundendatenbank angewiesen. Veraltete oder gar manipulierte Datenbestände können folgenschwere Schäden nach sich ziehen. Erhält

[23] BSI Glossar 2009 – Grundwerte der Informationssicherheit.
[24] BSI Glossar 2009 – Vertraulichkeit.
[25] BSI Glossar 2009 – Integrität.

beispielsweise ein Kunde ein Angebot mit falschen Konditionen, kann dies nicht nur zum Verlust des aktuellen Auftrags, sondern auch zum Ende der Geschäftsbeziehung führen. Da Teile einer Datenbank zum Zweck der Abfragebeschleunigung im Arbeitsspeicher vorgehalten werden, ist es einem Angreifer auf Hypervisor-Ebene möglich die Daten aus dem Arbeitsspeicher auszulesen oder zu manipulieren. Wird beispielsweise das virtuelle System temporär auf einen speziell präparierten Hypervisor übertragen, hat ein Systemadministrator mit bösen Absichten freien und unbemerkten Zugriff auf den Datenbestand und Einfluss auf alle Operationen.

2.3.4 Datenverfügbarkeit

Die Verfügbarkeit ist gewährleistet, wenn die Daten, Leistungen oder Systeme dem Benutzer wie gewünscht zur Verfügung stehen.[26] Die Verfügbarkeit kann als Prozentwert der durchschnittlichen Verfügbarkeit über ein Jahr angegeben werden. Dabei gilt es zwar eine Verfügbarkeit von 100%, also keine Ausfallzeiten, anzustreben, Werte zwischen 99,5 %, das entspricht 43,8 Stunden Nichtverfügbarkeit, und 99,99 % (0,53 Stunden Nichtverfügbarkeit) entsprechen aber eher der Realität. Amazon sieht beispielsweise eine Verfügbarkeit von 99,95 % für die Amazon Elastic Compute Cloud vor, wobei es sich aber um keine vertragliche Zusage handelt.[27] Der Rechenzentrumsbetreiber Hetzner hingegen garantiert eine Verfügbarkeit von 99 % in seinen allgemeinen Geschäftsbedingungen.[28]

[26] BSI Glossar 2009 – Verfügbarkeit.
[27] Amazon Web Services Inc. – EC2 Überblick. Online im Internet: URL: http://aws.amazon.com/de/ec2/ (Abgerufen am 9.2.2013).
[28] Hetzner Online AG – AGB. Online im Internet: URL: http://hetzner.de/hosting/legal/agb (Abgerufen am 9.2.2013).

3 Trusted Cloud Computing Platform

Die im Folgenden beschriebene *Trusted Cloud Computing Platform* (TCCP) basiert auf dem Paper *Towards Trusted Cloud Computing* von Santos et al. und beschreibt wie ein Infrastructure-as-a-Service Anbieter seine Infrastruktur erweitern muss, um die Sicherheit der betriebenen VMs zu erhöhen.

3.1 Voraussetzungen

Die TCCP setzt auf der konventionellen Virtualisierungsinfrastruktur auf. Das heißt als Ausgangssituation wird ein Rechenzentrum mit mehreren Hypervisors zugrunde gelegt. Die Rechenleistung der Hypervisors wird den Nutzern in Form von virtuellen Maschinen bereitgestellt. Ein Nutzer hat dabei keinen Einfluss auf den Hypervisor, auf dem seine VM ausgeführt wird. Außerdem wird vorausgesetzt, dass die Abschottungsmechanismen des Hypervisors den Zugriff zwischen den VMs ausschließen. Für die Realisierung wird ein TPM eingesetzt, das für Authentifizierungszwecke und die Vertrauenswürdigkeitsprüfungen (des Hypervisors) verwendet wird.

3.2 Infrastruktur

Die wesentliche Aufgabe einer Trusted Computing Platform ist der Start eines Systems, bei dem eine Protokollierung der Zustände und eine anschließende Verifizierung der Vertrauenswürdigkeit des aktuellen Zustands durchgeführt wird. Diese Funktionalität nutzt die TCCP, um einen vertrauenswürdigen Hypervisor zu starten. Voraussetzung für den sicheren Betrieb ist die gezielte Unterbindung von Zugriffen durch den Systemadministrator auf sensible Systemfunktionen. Beispiele hierfür sind die Verhinderung von Zugriffen auf den Arbeitsspeicher, die Festplatten der VMs oder die ausgeführten CPU-Operationen.

Die vertrauenswürdigen Hypervisors werden von einem sogenannten Trusted Coordinator (TC) verwaltet, welcher per Remote Attestation die Vertrauenswürdigkeit der Hypervisors überprüft. Damit der TC nicht von den Mitarbeitern des Rechenzentrums manipuliert werden kann, sollte dieser durch eine vertrauenswürdige – vom Rechenzentrum technisch und wirtschaftlich unabhängige – External Trusted Entity (ETE) verwaltet werden.

3.3 Verwaltung

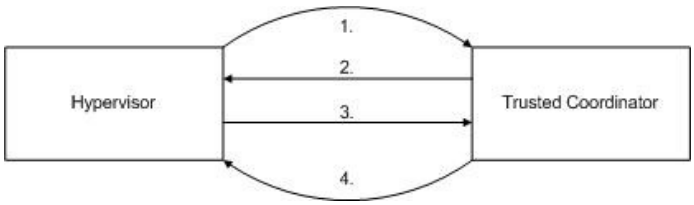
3.3.1 Hypervisor Management

Nur die Hypervisors in der Datenbank des TC sind als vertrauenswürdig einzustufen. Damit ein Hypervisor sich bei dem TC registrieren kann, ist folgendes Protokoll vereinbart:

Zuerst sendet der Hypervisor ein Challenge n_N an den TC. Dieser Antwortet mit seinem Platform Configuration Register PCR_{TC}, welches mit dem privaten Endorsment Key des TC (EK_{TC}^{P}) verschlüsselt ist und seinerseits mit einem Challenge n_{TC}. Durch die Verschlüsselung authentifiziert sich der TC gegenüber dem Hypervisor und mit dem PCR_{TC} kann der Hypervisor die Vertrauenswürdigkeit des Zustandes des TC überprüfen. Anschließend generiert der Hypervisor ein RSA-Schlüsselpaar mit dem privaten Schüssel TK_N^p und dem öffentlichen Schlüssel TK_N^P. Dem TC wird folgende Nachricht gesendet:

1. n_{TC} als Beweise für die Entschlüsselung
2. Das Platform Configuration Register PCR_N zur Beschreibung des Zustands
3. Der neu generierte öffentliche Schlüssel TK_N^P

Dabei sind 1. und 2. gemeinsam mit dem privaten Endorsment Key EK_N^P verschlüsselt, die Nachricht wiederum mit dem öffentlichen Schlüssel TK_{TC}^P. Stuft der TC den Hypervisor als vertrauenswürdig ein, wird der Schlüssel TK_N^P in die Datenbank aufgenommen und eine Bestätigungsnachricht, mit dem öffentlichen Schlüssel TK_N^P verschlüsselt, an den nun vertrauenswürdigen Hypervisor geschickt. Die Abbildung 2 stellt den Ablauf schematisch dar.



1. n_N
2. $\{PCR_{TC}, n_N\}_{EK_{TC}^P}, n_{TC}$
3. $\left\{\{PCR_N, n_{TC}\}_{EK_N^P}, TK_N^P\right\}_{TK_{TC}^P}$
4. $\{accepted\}_{TK_N^P}$

Abb. 2: Schema der Autorisierung zum Trusted Hypervisor.[29]

3.3.2 Virtual Machine Management

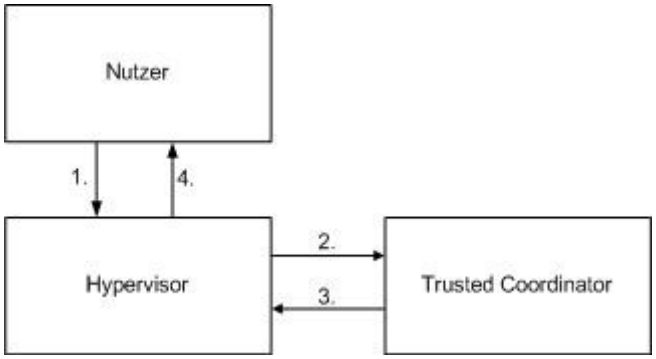
Vor dem Start einer VM ist noch nicht festgelegt auf welchem Hypervisor sie ausgeführt wird. Dies entscheidet sich erst zum Zeitpunkt der konkreten Start-Anforderung und ist von der Auslastung der Hypervisor im Rechenzentrum abhängig.

Der Nutzer vertraut darauf, dass der TC die Ausführung der VM auf einem vertrauenswürdigen Hypervisor sicherstellt. Bevor der Nutzer seine VM an das Rechenzentrum übermittelt, wird ein Session-Schlüssel K_{VM} erstellt, der für die symmetrische Verschlüsselung der VM für die Übertragung genutzt wird. Die VM wird zusammen mit ihrem Hashwert für die Integritätsprüfung verschlüsselt. Für die Entschlüsselung wird der Session-Schlüssel zusammen mit einem Nonce[30] mit dem öffentlichen Schlüssel des TCs verschlüsselt. Beides wird an das Rechenzentrum gesendet, welches dann einen passenden – angeblich vertrauenswürdigen – Hypervisor auswählt, auf dem die VM ausgeführt werden soll.

Um die VM zu starten, muss der Hypervisor diese zunächst entschlüsseln. Dafür kontaktiert er den TC und sendet ihm den, mit seinem privaten Schlüssel TK_N^p verschlüsselten, Session-Schlüssel. Durch die Entschlüsselung mit dem öffentlichen Schlüssel des Hypervisors kann der TC implizit sicherstellen, dass der Hypervisor vertrauenswürdig ist. Anderenfalls hätte er dessen öffentlichen Schlüssel nicht in seiner Datenbank. Der TC kann nun den Session-Schlüssel mit seinem privaten Schlüssel entschlüsseln. Anschließend übermittelt er diesen verschlüsselt mit dem öffentlichen Schüssel (TK_N^P) an den Hypervisor. Dadurch wird sichergestellt, dass nur ein

[29] Eigene Abbildung, in Anlehnung an Santos et al. 2009.
[30] Abk.: used only once oder number used once.

vertrauenswürdiger Hypervisor den Session-Schlüssel K_{VM} erhält, die VM entschlüsseln und anschließend starten kann. In der Abbildung 3 wird der Ablauf schematisch dargestellt.



1. $\{\alpha, \#\alpha\}_{K_{VM}}\{n_U, K_{VM}\}_{TK_{TC}^P}$
2. $\left\{\left\{\{n_U, K_{VM}\}_{TK_{TC}^P}, n_N\right\}_{TK_N^P}, N\right\}_{TK_{TC}^P}$
3. $\left\{\{n_N, n_U, K_{VM}\}_{TK_N^P}\right\}_{TK_{TC}^P}$
4. $\{n_N, N\}_{K_{VM}}$

Abb. 3: Schema der sicheren Datenübermittlung auf einen Trusted Hypervisor.[31]

Die Migration von einem Trusted Hypervisor auf einen anderen erfordert zusätzliche Kommunikation. Der vertrauenswürdige Quell-Hypervisor überprüft mit Hilfe des TC die Vertrauenswürdigkeit des Ziel-Hypervisors. Ist diese sichergestellt, übermittelt der Quell-Hypervisor einen Session-Schlüssel, mit dem die VM vor der Übertragung verschlüsselt wird. Bevor der Ziel-Hypervisor den Session-Schlüssel und einen Datentransfer akzeptiert, überprüft auch er mittels des TCs, ob der Quell-Hypervisor vertrauenswürdig ist. Erst wenn die Vertrauenswürdigkeit bestätigt ist, sendet der Ziel-Hypervisor eine Bestätigung an den Quell-Hypervisor, woraufhin dieser den Datentransfer der verschlüsselten VM beginnen kann. Die gesamte Kommunikation wird mit den privaten und öffentlichen Schlüsseln der beteiligten Systeme gesichert.

[31] Eigene Abbildung, in Anlehnung an Santos et al. 2009.

3.4 Verbesserungen

Der Reihenfolge aus Abschnitt 2.3 folgend, soll das Protokoll als erstes hinsichtlich der Datenvertraulichkeit untersucht werden. Die vertraulichen Daten umfassen hier die gesamte VM vor, während und nach der Ausführung. Der Zeitraum vor der Ausführung betrifft alle Datenübertragungen, die bis zum Start der VM auf dem Trusted Hypervisor stattgefunden haben. Das Protokoll stellt sicher, dass die VM bei jeder Datenübertragung verschlüsselt ist. Die Vorgabe ein symmetrisches Verschlüsselungsverfahren einzusetzen, kommt der Ver- und Entschlüsselungsgeschwindigkeit zugute. Dem wesentlichen Nachteil der symmetrischen Verschlüsselung – die Verwendung desselben Schlüssels zur Ver- und Entschlüsselung sowie dessen Übertragung – wird mit einer asymmetrischen Verschlüsselung des Schlüssels selbst begegnet. Es kann davon ausgegangen werden, dass der Schlüssel keinem Angreifer von außen oder aus dem Rechenzentrum zugänglich wird. Dementsprechend hängt die Sicherheit im Wesentlichen vom eingesetzten Verfahren ab. Ausreichend sichere Verfahren wie zum Beispiel AES oder Triple-DES stehen zur Verfügung.

Während der Ausführung hängt die Vertraulichkeit der Daten im Wesentlichen vom Trusted Hypervisor ab. Es muss sichergestellt sein, dass der Trusted Hypervisor keine Möglichkeit bietet ein unverschlüsseltes Abbild der VM anzufertigen. Ebenso muss der Zugriff auf die unverschlüsselte VM oder auf den unverschlüsselten Inhalt des Arbeitsspeichers verhindert werden. Ist dies gewährleistet, muss nur noch sichergestellt werden, dass auch wirklich der Trusted Hypervisor ausgeführt wird. Hierfür wird das TPM eingesetzt, welches seit dem Start der physikalischen Maschine die Hashwerte der Systemzustände gesammelt hat. Gegenüber dem TC authentifiziert sich der Hypervisor als vertrauenswürdig und wird von diesem als Trusted Hypervisor zur Ausführung von VMs autorisiert. Die Autorisierung erfolgt allerdings nur, wenn der Systemzustand dem erwarteten, vertrauenswürdigen Zustand entspricht. Da die gesamte Kommunikation mit dem asymmetrischen Verschlüsselungsverfahren RSA verschlüsselt wird, kann sie als sicher eingestuft werden. Nach der Ausführung, also sobald die VM beendet ist, ist es Aufgabe des Trusted Hypervisors den Zugriff nur auf die verschlüsselte VM zu ermöglichen. Im besten Fall ist nur der sichere Transfer einer VM zu einem anderen Trusted Hypervisor oder einem externen vertrauenswürdigen System über definierte Schnittstellen möglich.

Zur Gewährleistung der Integrität wird ein Hashwert der VM bei der Datenübertragung mitgeliefert. Damit kann der Empfänger nach der Entschlüsselung feststellen, ob die VM bei der Übertragung verändert wurde. Dadurch ist allerdings nicht festgestellbar, ob die VM bei der Übertragung abgefangen, dann entschlüsselt, manipuliert und wieder verschlüsselt wurde. Der Angreifer würde in diesem Falle auch einen neuen Hashwert erzeugen und diesen mitliefern. Die unabhängige und verschlüsselte Übertragung des Hashwertes zum TC würde dieses Problem beheben. Das grundsätzliche Verfahren, die Integrität eines Abbildes durch dessen Hashwerte zu überprüfen, ist gängige Praxis und kann als sicher eingestuft werden. Ein Nonce wird an den entscheidenden Stellen der Kommunikation eingesetzt, um beispielsweise Reply-Angriffe zu verhindern. Die Authentizität der Kommunikation wird bereits implizit durch die asymmetrische Verschlüsselung sichergestellt, sodass ist ein zusätzlicher Message Authentication Code nicht notwendig ist.

Auf die Datenverfügbarkeit hat das vorgestellte Protokoll keinen Einfluss. Dennoch ist dieses Sicherheitsziel für ein Unternehmen wie die Beispiel GmbH von großer Bedeutung. Abhängig von der notwendigen Verfügbarkeit muss ein geeigneter Anbieter ausgewählt werden. Ein möglicher Datenverlust, wie bei der Amazon Elastic Compute Cloud bereits vorgekommen, ist nicht akzeptabel.[32] Auch längere Ausfallzeiten und die damit verbundende Nichtverfügbarkeit von Diensten ist für Unternehmen kaum hinnehmbar. Diese Bedingungen müssen abhängig vom Einzelfall untersucht und bewertet werden.

[32] Blodget – Amazon's Cloud Crash Disaster Permanently Destroyed Many Customers' Data. Online im Internet: URL: http://businessinsider.com/amazon-lost-data-2011-4 (Abgerufen am 9.2.2013).

4 Fazit und Ausblick

Das vorgestellte Protokoll nutzt die Mechanismen des Trusted Computing, um eine sichere Kommunikation zwischen allen Beteiligten zu gewährleisten und um zu verifizieren, ob ein Hypervisor als vertrauenswürdig einzustufen ist oder nicht. Der Trusted Coordinator spielt dabei eine zentrale Rolle, da sich die Nutzer darauf verlassen, dass er zuverlässig die vertrauenswürdigen Hypervisors benennen kann. Eine solche Infrastruktur ist als Ganzes nicht ungewöhnlich und hat sich in Form von Zertifizierungsstellen bereits im eCommerce bewährt. Denkbar ist, dass die heutigen Zertifizierungsstellen die Aufgabe des Trusted Coordinators übernehmen können. Die Verschlüsselung der gesamten Kommunikation lässt sich auf die bewährten und als sicher eingestuften Standardverfahren der Kryptographie zurückführen. Die Kombination von symmetrischen und asymmetrischen Verfahren erlaubt es die Vorteile beider zu nutzen, ohne das daraus Nachteile für den Nutzer entstehen. Die Ausgangsfrage „Kann die Sicherheit von Cloud Anwendungen durch Trusted Computing erhöhrt werden?“ kann zusammenfassend bejahend beantwortet werden. Insbesondere die Sicherheitsziele Vertraulichkeit und Integrität können so erreicht werden. Die Datenverfügbarkeit hingegen ist, wie sich herausgestellt hat, unabhängig vom Trusted Computing. Dennoch müssen Unternehmen bei der Auslagerung in die Cloud besondere Maßnahmen ergreifen, da die Verfügbarkeit dann von zusätzlichen Komponenten wie dem Internet abhängig ist. Neben der noch offenen Realisierung und dem Sammeln von Praxiserfahrungen gibt es bereits eine Weiterentwicklung auf konzeptioneller Ebene. Das Paper *Policy-Sealed Data: A New Abstraction for Building Trusted Cloud Services* stellt eine Erweiterung der Trusted Cloud Computing Platform vor, welche zum einen die Geschwindigkeitsbegrenzung des TPM überwindet und zum anderen eine feingranulare Definition von Ausführungsrichtlinien erlaubt.[33] Dadurch erhalten Nutzer die Möglichkeit die Ausführungsbedingungen Ihrer VM – beispielsweise den Ausführungsort – verbindlich zu bestimmen.

Der rasante Fortschritt im Cloud Computing und das immer stärker werdende Interesse von Unternehmen ihre Rechenleistung auszulagern, wird auf diesem Gebiet noch weitere Innovationen hervorbringen.

[33] Siehe Santos et al 2012.

Quellenverzeichnis

A Secure and Reliable Bootstrap Architecture
W. A. Arbaugh, D. J. Faber, J. M. Smith
University of Pennsylvania, 1997

Glossar und Begriffsdefinitionen
Bundesamt für Sicherheit in der Informationstechnik
Stand: 11. EL Stand 2009

Praxishandbuch VMware vSphere 5
Ralph Göpel
2. Ausgabe, 2012, O'Reilly Germany

Virtualization: A Manager's Guide: A Manager's Guide
Dan Kusnetzky
O'Reilly Media, Inc., 2011

Trusted Computing Systeme – Konzepte und Anforderungen
Thomas Müller
Korrigierter Nachdruck 2008, Springer-Verlag, Berlin Heidelberg

Towards Trusted Cloud Computing
Nuno Santos, Krishna P. Gummadi and Rodrigo Rodrigues
Proceedings of HotCloud, San Diego, CA, USA, June 2009

Policy-Sealed Data: A New Abstraction for Building Trusted Cloud Services
Nuno Santos, Rodrigo Rodrigues, Krishna P. Gummadi and Stefan Saroiu
Proceedings of USENIX Security, Bellevue, WS, USA, August, 2012

Dyad: A System for Using Physically Secure Coprocessors
J. D. Tygar, B. Yee
Technical Report CMU-CS91-140R, Carnegie Mellon University, May 1991

Internet Security Threat Report, Volume 17
Symantec, www.symantec.com/threatreport/ (Abgerufen am 9.2.2013)

ISO/IEC 11889-2:2009