

Einführung in die IT-Sicherheit

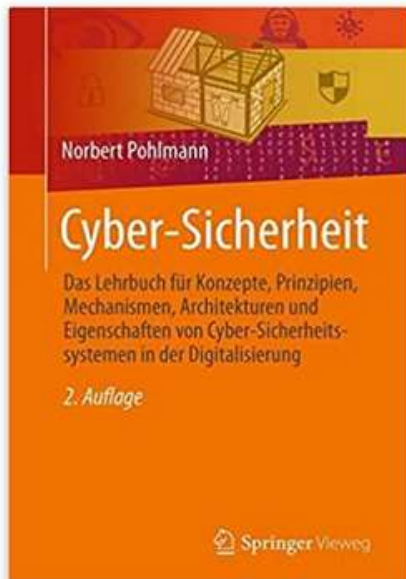
Prof. Dr. Jean-Pierre Seifert

jpseifert@sec.t-labs.tu-berlin.de

<http://www.sec.t-labs.tu-berlin.de/>



Literatur



[Dieses Bild anzeigen](#)

Cyber-Sicherheit: Das Lehrbuch für Konzepte, Prinzipien, Mechanismen, Architekturen und Eigenschaften von Cyber-Sicherheitssystemen in der Digitalisierung Taschenbuch – 23. Mai 2022

von [Norbert Pohlmann](#) (Autor)

[Alle Formate und Editionen anzeigen](#)

Taschenbuch

32,99 € ✓prime

1 Neu ab 32,99 €

Dieses Lehrbuch gibt Ihnen einen Überblick über die Themen der IT-Sicherheit Die digitale Transformation eröffnet viele neue Möglichkeiten, den dadurch lassen sich Geschäftsmodelle und Verwaltungsprozesse radikal verändern. Aber mit fortschreitender Digitalisierung nimmt jedoch die Komplexität der IT-Systeme- und Infrastrukturen zu. Zudem werden die Methoden der professionellen Angreifer ausgefeilter und die Angriffsziele kontinuierlich lukrativer, insgesamt führt dies bei Unternehmen und der Gesellschaft zu hohen Schäden. Für eine erfolgreiche Zukunft unserer Gesellschaft ist es daher entscheidend, diesen gestiegenen Risiken entgegenzuwirken und eine sichere sowie vertrauenswürdige IT zu gestalten. Von daher ist es notwendig, dass mit den wachsenden Herausforderungen auch neue Entwicklungen und Prozessen in der Cyber-Sicherheit einhergehen. Was sich hier getan hat können Sie in der 2. Auflage des Lehrbuchs ‚Cyber-Sicherheit‘, von Prof. Norbert Pohlmann, nachlesen. Denn in der Überarbeitung der sehr erfolgreichen Erst-Auflage wurden die bestehenden Kapitel ergänzt und aktualisiert sowie zusätzlich für neue Themen weitere Kapitel hinzugefügt. Aber auch Lehrmaterialien, wie 19 komplette Vorlesungen und Überbungen auf den Webseiten wurden angepasst und erweitert.

Auf insgesamt 746 Seiten bietet Informatikprofessor Norbert Pohlmann grundlegendes Wissen über die Cyber-Sicherheit und geht bei innovativen Themen, wie Self Sovereign Identity oder dem Vertrauenswürdigkeits-Modell, detailliert in die Tiefe. Dabei ist dem Autor wichtig, nicht nur theoretisches Fachwissen zu vermitteln, sondern auch den Leser in die Lage zu versetzen, die Cyber-Sicherheit aus der anwendungsorientierten Perspektive zu betrachten.

Lernen Sie mithilfe dieses Lehrbuchs mehr über Mechanismen, Prinzipien, Konzepte und Eigenschaften von Cyber-Sicherheitssystemen. So sind Sie in der Lage, die Sicherheit und Vertrauenswürdigkeit von IT-Lösungen zu beurteilen.

0. Organisation

□ **Lerninhalte:**

Den Studierenden werden grundlegende Kenntnisse bezüglich der folgenden Themen vermittelt:

- Begriffe, Definitionen, Zielsetzung und Motivation der IT-Sicherheit
- Kryptographie: Symmetrische und asymmetrische Verschlüsselung
- Identifikation und Authentifikation
- Entwurf sicherer Systeme + Access Control + Principle of Least Privilege
- Netzwerksicherheit
- Schwachstellen in Soft- und Hardware
- Security Testing

Trusted Computing

→ **Inhalt**

- ❑ Ziele und Ergebnisse der Vorlesung
- ❑ Einleitung
- ❑ IT-Sicherheitsarchitektur
- ❑ Trusted Computing-Funktion
- ❑ Trusted Network Connect - TNC
- ❑ Zusammenfassung

Trusted Computing

→ **Inhalt**

- ❑ Ziele und Ergebnisse der Vorlesung
- ❑ Einleitung
- ❑ IT-Sicherheitsarchitektur
- ❑ Trusted Computing-Funktion
- ❑ Trusted Network Connect - TNC
- ❑ Zusammenfassung

Ziele und Ergebnisse der Vorlesung

→ **Trusted Computing**

- ❑ Gutes Verständnis über die **IT-Sicherheitsarchitektur** und **IT-Sicherheitsprinzipien** von Trusted Computing erlangen.
- ❑ Erlangen der Kenntnisse über die *TPM Schlüsselhierarchie*, **Authenticated Boot**, **Attestation**, **Binding**, **Sealing** und *Trusted Network Connect (TNC)*
 - → etwas später in VL.
- ❑ Gutes Verständnis zu verschiedenen **Kernelarchitekturen** erlangen.
- ❑ Gutes Verständnis über praktische Anwendungen mit Hilfe der IT-Sicherheitsplattform **Turaya**.

Trusted Computing

→ **Inhalt**

- ❑ Ziele und Ergebnisse der Vorlesung
- ❑ **Einleitung**
- ❑ IT-Sicherheitsarchitektur
- ❑ Trusted Computing-Funktion
- ❑ Trusted Network Connect - TNC
- ❑ Zusammenfassung

Trusted Computing

→ Definition von Vertrauenswürdigkeit

- Ein IT-System ist vertrauenswürdig, wenn es sich immer in der erwarteten Weise für den beabsichtigten Zweck verhält.

- Wie kann das überprüft werden?
 - Der Nutzer macht seine **eigenen Erfahrungen** mit der erwarteten Weise für den beabsichtigten Zweck des IT-Systems
oder
 - Der Nutzer **vertraut jemandem**, der Referenzen für die erwartete Weise des beabsichtigten Zweck des IT-Systems bereitstellt
(Vertrauenspersonen, Reputationssysteme, Zertifizierungen, ...)
oder
 - **Mit Trusted Computing Funktionen**

Trusted Computing

→ Idee

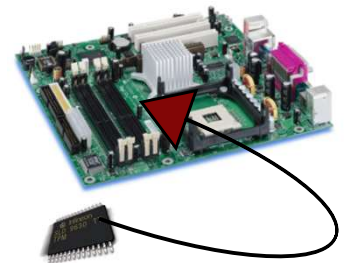
- ❑ **Trusted Computing Group (TCG):**
Industriekonsortium bestehend aus den führenden IT-Firmen (Hewlett-Packard, IBM, Intel, AMD, Microsoft, Sony, Sun, Infineon, STM, ...)
- ❑ **Grundmotivation**
 - Entwicklung **offener Spezifikationen** für **vertrauenswürdige IT-Systeme** (Server, PC, eingebettet, usw.)
 - Sicherheit verteilter Anwendungen mit wirtschaftlich vertretbarem Aufwand verbessern
 - Keine massive Veränderung existierender Hard- bzw. Software
- ❑ **Hauptidee**
 - Manipulationssichere Komponente in Hardware (sicherer als Software) → **Stärkung gegen Software-basierte Angriffe**
 - Sicherheit des IT-Systems reduziert auf die Sicherheit eines Moduls
 - Integrität und Authentizität eines IT-Systems zuverlässig überprüfbar, auch aus der Distanz

Trusted Computing

→ Funktionen (1/2)

❑ **Trusted Platform Modules (TPM)**

- Sicherer Zufallsgenerator (sichere kryptographische Schlüssel)
- Kryptographische Funktionen: Signatur (RSA), Hash (SHA-X)
- Erzeugung verschiedener kryptographischer Schlüssel
- **Platform Configuration Register (PCR)**
→ **Zur Speicherung der Systemkonfiguration**



TPM

❑ **Sicherer Speicher**

- **Erzeugung sicherer kryptographischer Schlüssel** und
- **sichere Speicherung von Schlüssel** im Hardware-Sicherheitsmodul (TPM)

❑ **Sealing (versiegeln)**

- **Kryptographische Schlüssel** können an das **IT-System** und/oder eine bestimmte **Softwarekonfiguration** gebunden werden
→ Schutz vor Manipulationen des Betriebssystems

Trusted Computing

→ Funktionen (2/2)

❑ (Remote) Attestation

- Aktuelle Systemkonfiguration des IT-Systems wird gemessen
- Erkennung manipulierter IT-Systeme (Verteilte Systeme, Web-S.)
- Kommunikation nur mit vertrauenswürdigen IT-Systemen

❑ Access Control

- Durchsetzung von Zugriffsregeln in einem Netzwerk mit unbekannten IT-Systemen (TNC)

❑ Überprüfbares Booten

- Systemkonfiguration kann überprüft werden, z.B. mittels eines persönlichen Gerätes (Smartcard, USB-Stick, Handy, ...)

❑ Verbreitung von TPMs

- Da IT-Systeme wie Notebooks fast ausschließlich „**Microsoft Ready**“ sind, ist ein **TPM** auf den meisten der **IT-Systeme** verfügbar.



TPM

Digitalisierung

→ Herausforderungen

- **Das Software-Problem**
- **Malware auf zu vielen IT-Systemen**

Durch eine starke Isolierung den Schaden begrenzen

Cyber-Sicherheitssysteme

→ **Reaktive Cyber-Sicherheitssysteme**

- Bei den heutigen **reaktiven Cyber-Sicherheitssystemen**, wie Anti-Spam-, Anti-Malware-, Intrusion-Detection-Systemen, ist die **grundsätzliche Idee**, so **gut** und **schnell** wie möglich **Cyber-Angriffe zu erkennen**.
- Das bedeutet, wenn die Cyber-Sicherheitslösungen einen Angriff durch eine entsprechende **Angriffssignatur** oder eine **Anomalie** erkennen, dann wird versucht, das IT-System so schnell wie möglich zu schützen, um den Schaden zu reduzieren.

„Airbag-Methode“

Wenn's passiert, soll es weniger wehtun!



Cyber-Sicherheitssysteme

→ Proaktive Cyber-Sicherheitssysteme

- Für die zunehmende **Vielfalt und Komplexität** der IT-Endgeräte, IoT-Geräte, Netzkomponenten und IT-Infrastrukturen werden deutlich **verlässlichere, robustere** und **wirkungsvollere Cyber-Sicherheitskonzepte** benötigt.
- Daher ist es sinnvoll, weniger reaktive und mehr **moderne proaktive Cyber-Sicherheitssysteme** zu verwenden.
- Proaktiven Cyber-Sicherheitssysteme arbeiten mit einem kleinen **Sicherheitskern** (sichere Betriebssysteme) und **Virtualisierung**, können Software messbar machen und mit einer **starken Isolation** Anwendungen mit ihren Daten separieren und so nachhaltige und angemessene Cyber-Sicherheit bieten.
- Für proaktive Cyber-Sicherheitssysteme muss die Softwarearchitektur der IT-Endgeräte allerdings grundlegend anders aufgebaut sein als bisher.

„ESP-Strategie“

Verhindern, dass ein „Auto“ überhaupt ins Schleudern kommt



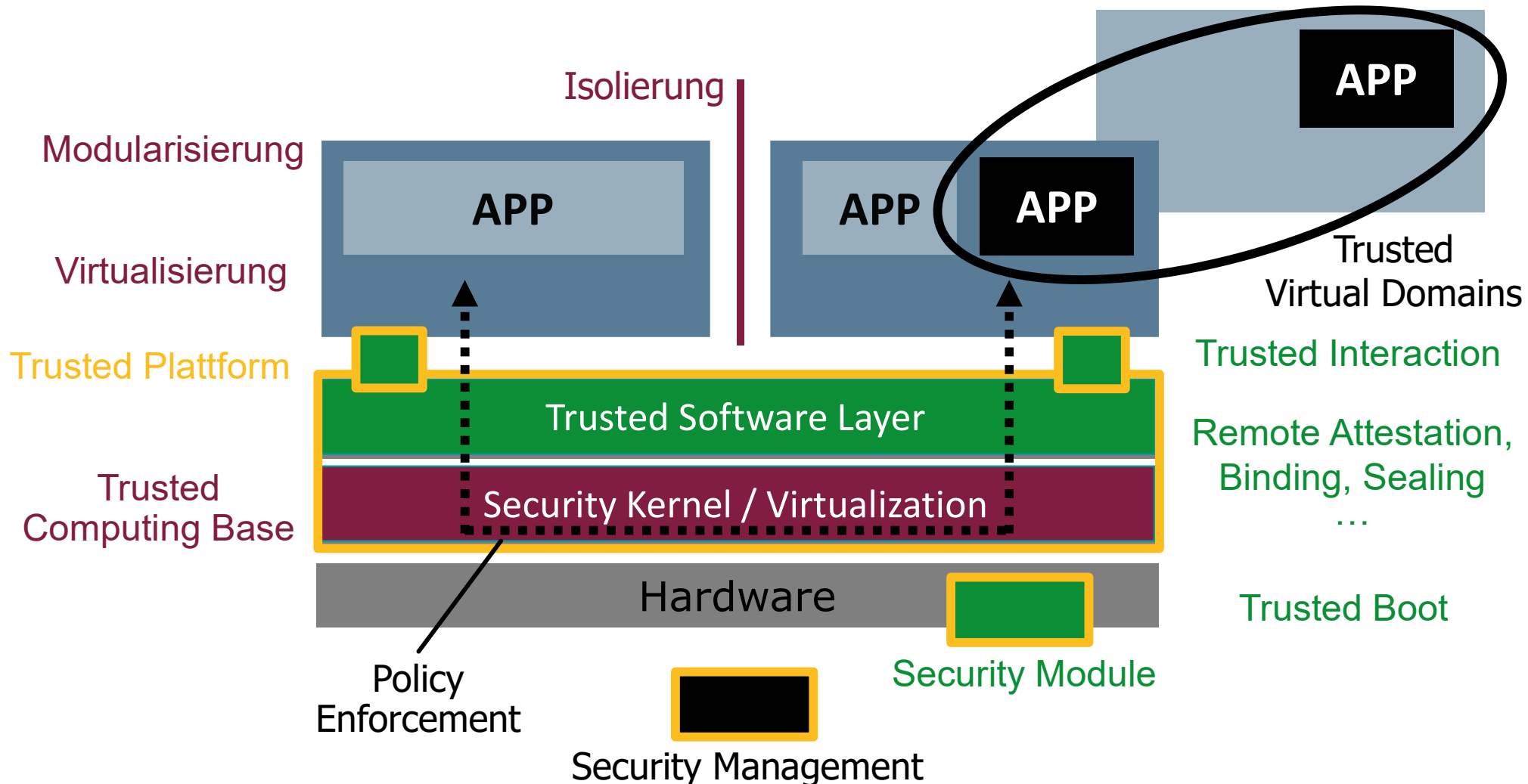
IT-Sicherheitsarchitektur

→ IT-Sicherheitsprinzipien

Robustness/Modularity

Trusted Process

Integritätsprüfung



TC-Sicherheitsaspekte

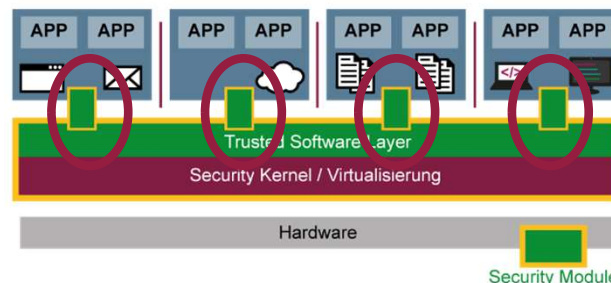
→ **Virtualisierung**

- Ein wichtiger IT-Sicherheitsaspekt ist die Virtualisierung auf dem Endgerät.
- Der Vorteil von Virtualisierung besteht darin, dass **auf tretende Fehler** (Schwachstelle, Malware, ...) im Prinzip in einer virtuellen Maschine in einem **abgeschlossenen Bereich begrenzt bleibt** und nicht eine andere virtuelle Maschine infizieren kann.
- Es ist auch sehr einfach möglich, die verschiedenen virtuellen Maschinen wieder in einen **stabilen Urzustand** zu versetzen und von da aus neu zu starten.

TC-Sicherheitsaspekte

→ **Isolierung**

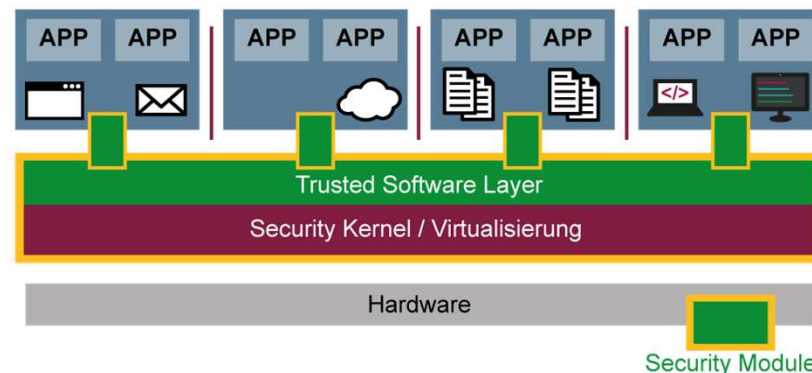
- Der IT-Sicherheitsaspekt **Isolierung** sorgt dafür, dass die **virtuellen Maschinen** zusätzlich weiter stark isoliert und sicher getrennt voneinander laufen und **sich nicht gegenseitig beeinflussen** können.
- Aus diesem Grund haben z.B. Schwachstellen und Malware in einer isolierten virtuellen Maschine **keinen Einfluss auf die anderen virtuellen Maschinen** hat.
- Eine solche stark isolierte virtuelle Maschine wird im Bereich von TC auch Compartment genannt.
- Methoden der Isolierung:
 - **Enforcement der Kommunikation**
 - Individuelle **Verschlüsselung der Daten** in den virtuellen Maschinen



TC-Sicherheitsaspekte

→ Modularisierung

- Der IT-Sicherheitsaspekt der Modularisierung ist eine **Möglichkeit**, **Anwendungen**, die **zusammen** gehören, in einer virtuellen Maschine laufen zu lassen und Anwendungen, die **getrennt** sein sollten, in **verschiedenen virtuellen Maschinen** zu positionieren.
- Dieser IT-Sicherheitsaspekt offeriert einen interessanten **Gestaltungsspielraum**, mit dem eine sehr hohe Cyber-Sicherheit erzielt werden kann, weil für **verschiedene IT-Sicherheitslevel** von Anwendungen unterschiedliche virtuelle Maschinen genutzt werden können.



- **Eine Beispiel ist:**
Das **Office-Paket** läuft in einer virtuellen Maschine, das **Design-Paket** für die **Business-Anwendung** in einer anderen und der **Browser** hat auch eine separate virtuelle Maschine.

TC-Sicherheitsaspekte

→ **Trusted Boot/Authenticated Boot**

- Mithilfe von **Trusted Boot** oder **Authenticated Boot** kann dafür gesorgt werden, dass ein IT-System nur in einem **definierten vertrauenswürdigen Zustand** aktiv wird.

TC-Sicherheitsaspekte

→ Remote Attestation

- **Remote Attestation** gibt die Möglichkeit, die **Vertrauenswürdigkeit** von **anderen, auch fremden IT-Systemen zu messen**, bevor eine Interaktion mit diesem IT-System begonnen wird.

TC-Sicherheitsaspekte

→ **Binding/Sealing**

- Binding und Sealing sind weitere Trusted Computing-Funktionen, mit denen moderne IT-Sicherheitssysteme intelligent und vertrauenswürdig umgesetzt werden können.
- Bei **Binding** werden **verschlüsselte Daten an ein TPM gebunden**.
- Bei **Sealing** werden **verschlüsselte Daten an die Software- und Hardware-Konfiguration eines IT-Systems** sowie an ein TPM **gebunden**.

Trusted Computing

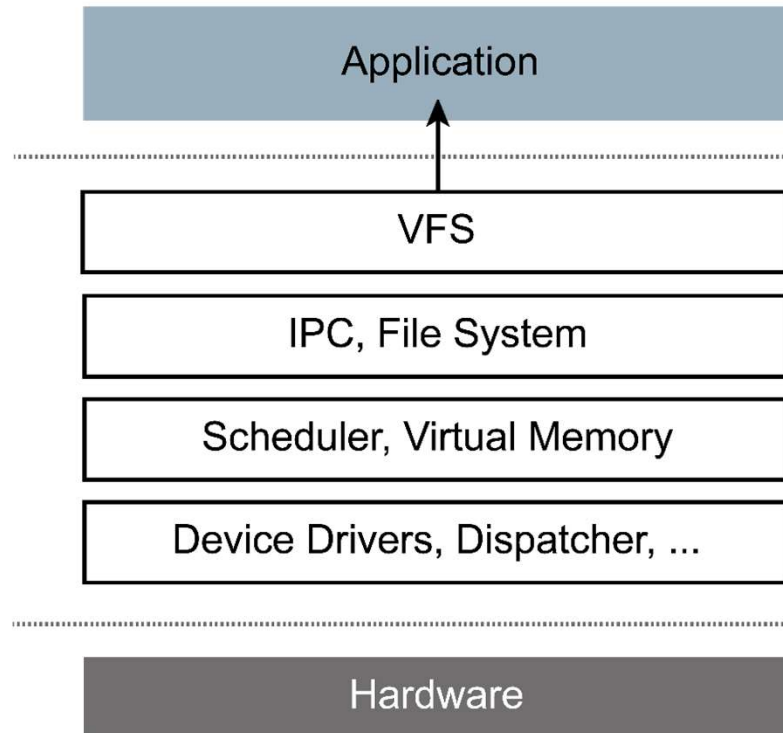
→ **Inhalt**

- ❑ Ziele und Ergebnisse der Vorlesung
- ❑ Einleitung
- ❑ IT-Sicherheitsarchitektur
- ❑ Trusted Computing-Funktion
- ❑ Trusted Network Connect - TNC
- ❑ Zusammenfassung

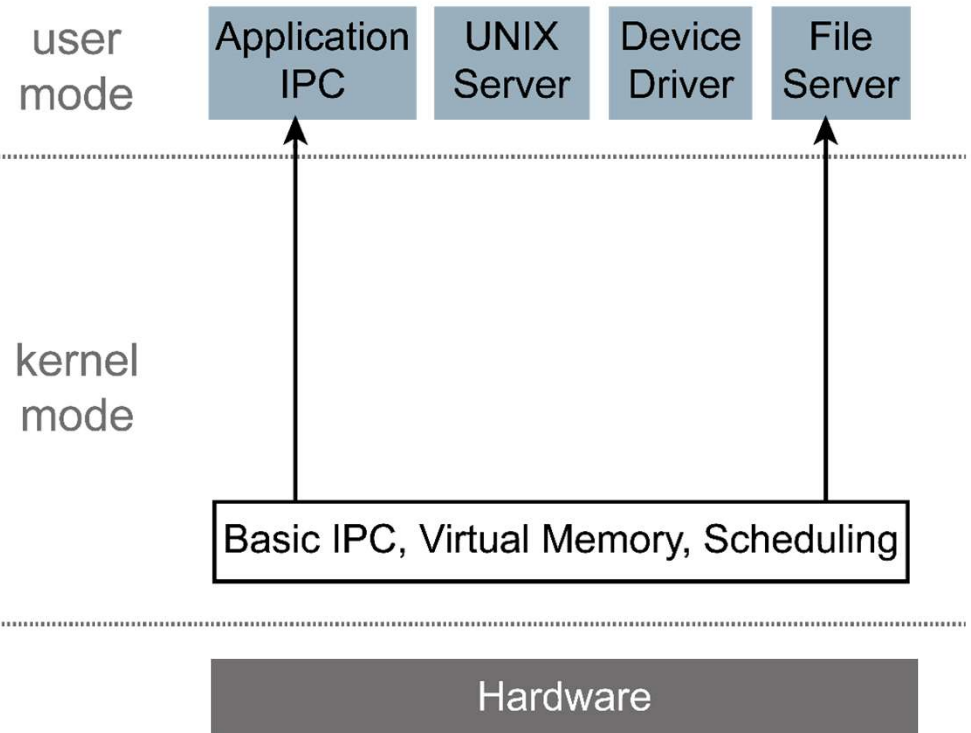
IT-Sicherheitsarchitektur

→ Kernelarchitekturen (1/3)

Monolithic Kernel based Operating System



Microkernel based Operating System



IT-Sicherheitsarchitektur

→ **Kernelarchitekturen (2/3)**

❑ **Vorteile** eines monolithischen Kernels:

- Lange etabliert
- Gute Performance

❑ **Nachteile** eines monolithischen Kernels:

- Alle Treiber vereint im Kernel-Space
- Geringere Flexibilität
- Höhere Komplexität
- Wenig robust
- Schlechte Sicherheitsmechanismen

IT-Sicherheitsarchitektur

→ **Kernelarchitekturen (3/3)**

❑ **Vorteile** eines Mikrokernels:

- Höhere Robustheit
- Höhere Modularität
- Höhere Flexibilität
- Höhere IT-Sicherheit (kontrollierbare Interprozesskommunikation)
- Weniger benötigter Speicherplatz

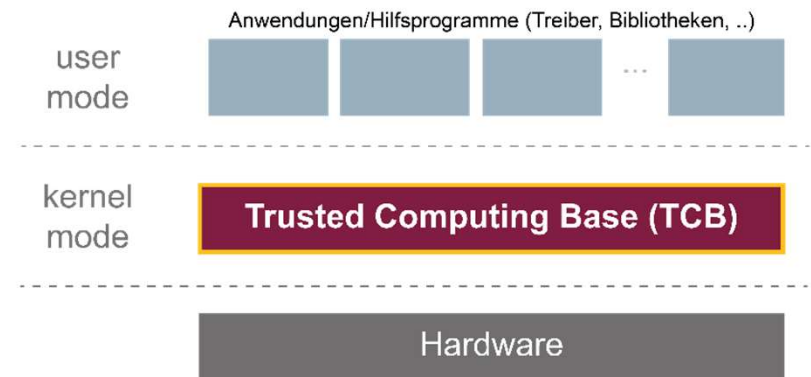
❑ **Nachteile** eines Mikrokernels:

- Weniger Leistung durch mehr Kommunikation zwischen den Prozessen

IT-Sicherheitsarchitektur

→ Trusted Computing Base (TCB) – 1/2

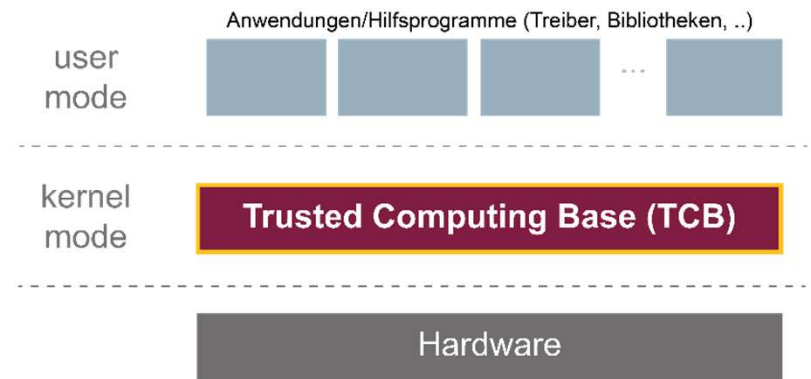
- Eine *Trusted Computing Base* dient als verlässliches **IT-Sicherheitsfundament**, um darauf weitere IT- und IT-Sicherheitskomponenten aufzubauen.
- Per Definition ist daher die „**Trusted Computing Base**“ der kritische Teil eines IT-Systems.
- Wenn im TCB eine **Schwachstelle** vorhanden ist, dann ist das ganze **IT-System** kompromittierbar.
- Wenn **außerhalb** der TCB eine Schwachstelle vorhanden ist, dann kann anhand einer **IT-Sicherheitspolicy** der **potenzielle Schaden** sehr eingeschränkt und **klar beschrieben** werden.



IT-Sicherheitsarchitektur

→ Trusted Computing Base (TCB) – 2/2

- ❑ Aus diesem Grund ist eine TCB **sehr sorgfältig designed und implementiert.**
- ❑ Eine auf einem Mikrokern (Security Kernel) basierende TCB hat ca. 20.000 Lines of Code und ist von daher eine sehr **vertrauenswürdige Basis**, die in der Regel auch schon semi-formal oder **formal verifiziert werden kann.**
- ❑ Mithilfe der formalen Beweisbarkeit wird eine **Sicherheitsevaluation auf hohem Niveau** möglich.
- ❑ Es gibt aber auch TCBs, die zum Beispiel aus einem sehr abgespeckten und speziell gehärteten Linux bestehen, das auch schon sehr viel vertrauenswürdiger sind als übliche Betriebssysteme.



Security Module

→ **Trusted Platform Module (TPM)**

- ❑ Das Security Module ist z.B. ein TPM mit kryptografischen Verfahren auf dem Level von Smartcard-Sicherheit, aber auch weiteren IT-Sicherheitsdiensten, wie die **Platform Configuration Register (PCR)**, die die sichere Speicherung und Überprüfung von Messdaten sicherstellt.
- ❑ Das TPM ist ein kleiner passiver Hardware-Sicherheitschip, der fest mit dem Mainboard verbunden ist.

Vorteile eines TPMs

- ❑ Die Hardware-Sicherheitsmodule bieten eine **sehr hohe IT-Sicherheit** bei **geringer Investitionssumme**, da ein TPM nicht mehr als ein Euro kostet.
- ❑ Die Hardware-Sicherheitsmodule sind schon auf den meisten IT-Systeme verfügbar, das heißt, die flächendeckende Einführung einer Sicherheitsplattform ist einfach! *(Alle IT-Systeme, die „Microsoft Ready“ sind, müssen ein TPM verbaut haben.)*
- ❑ Das **TPM** ist in eine **Sicherheitsinfrastruktur (PKI, ...)** eingebunden und daher einfach im Sicherheitsmanagement zu behandeln.

HSM: Trusted Platform Module

→ **Hardware-Sicherheitsanker**

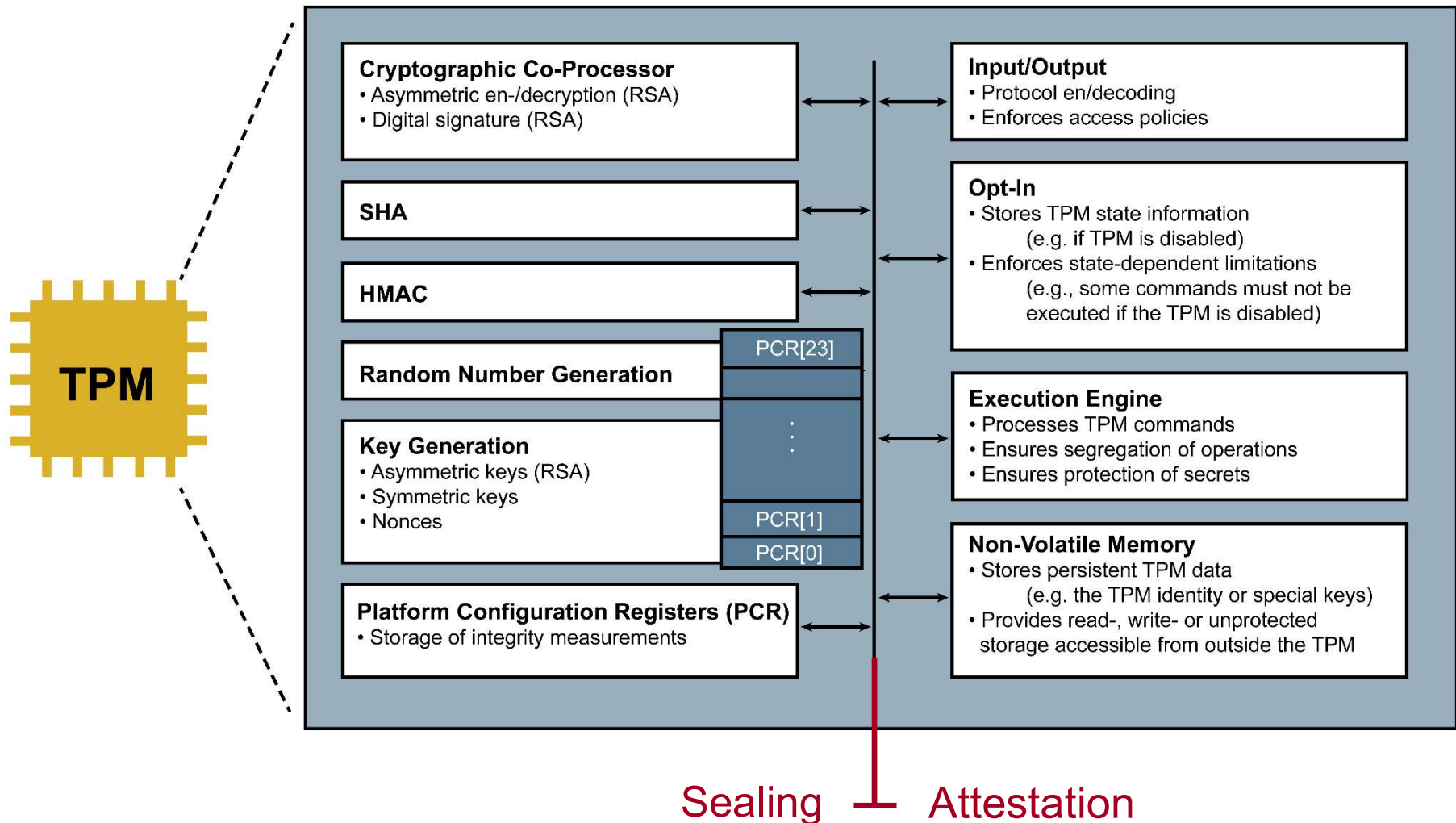
Basisfunktionen



HSM: Trusted Platform Module

→ Hardware-Sicherheitsanker

Basisfunktionen



IT-Sicherheitsarchitektur

→ Core Root of Trust for Measurement (CRTM)

- ❑ Eine **vertrauenswürdige Basis**, bzw. eine Trusted Platform, braucht eine Wurzel oder **Basis des Vertrauens**.
- ❑ Die Lösung bei der „Trusted Computing Group“ heißt „**Basis Root of Trust**“, die eine **Kette des Vertrauens** (Chain of Trust) bildet.
- ❑ Für normale IT-Systeme wurde von der Trusted Computing Group die Komponente „**Core Root of Trust for Measurement (CRTM)**“ spezifiziert.
- ❑ Der CRTM ist dabei eine Software, die einen Messvorgang über einzelne Systemzustände (Hard- und Software) außerhalb der Trusted Platform vertrauenswürdig durchführt und dann die Ergebnisse innerhalb der Trusted Platform in die Platform Configuration Register (PCR) des TPMs hinterlegt.
- ❑ Die Ausführung der CRTM Software beginnt mit dem Bootvorgang.
- ❑ Technisch wird die **CRTM Software** in der Regel in das **BIOS** des IT-Systems integriert.

→ Idee von transitiven Vertrauen

Entity E_0

CRTM

Core Root of
Trust for Measurement

"Transitives Vertrauen,,

- ❑ **Vertrauen ist transitiv** von E_0 nach E_1 nach E_2 bis E_n .
- ❑ Das Vertrauen von E_2 erfordert, dass E_0 und E_1 vertraut werden kann.
- ❑ Bei dem Konzept „Chain of Trust for Measurement“ ist das Ziel, das Vertrauen in Entität E_n zu gewinnen.
- ❑ Der Ablauf ist so: E_0 startet E_1 , E_1 startet E_2 usw.
- ❑ Um E_n zu vertrauen, muss E_{n-1} vertraut werden. Um E_{n-1} zu vertrauen, muss E_{n-2} vertraut werden usw.
- ❑ E_0 , E_1 bis E_n schaffen eine „**Vertrauenskette**“

→ **Authenticated Boot / Secure Boot**

■ **Authenticated Boot:**

- Systemzustände messen.
- Speicherung in den PCRs.
- Überprüfung der Integrität.

■ **Secure Boot:**

- Systemzustände messen.
- Überprüfung der Integrität.
- **Ggf. Bootvorgang stoppen.**

IT-Sicherheitsarchitektur

→ Identitäten (1/2)

□ **Endorsement Key (EK):**

- Eindeutige TPM-Identität (nicht migrierbar).
- RSA-Schlüsselpaar (im Herstellungsprozess erzeugt).
- Geheimer Schlüssel im TPM gespeichert.
- Öffentlicher Schlüssel ist datenschutzsensitiv.
- TPM-Hersteller verwaltet PKI.

□ **Endorsement Credential (EC):**

- Elektronisches Zertifikat vom TPM-Hersteller.
- Bestätigt ordnungsgemäße Erstellung und Einbettung des EK.
- Bestandteile: TPM-Herstellernamen, TPM-Modellnummer, TPM-Version, Öffentlicher Schlüssel des EK.

IT-Sicherheitsarchitektur

→ Identitäten (2/2)

■ Platform Identität (PI):

- Entspricht der TPM-Identität (EK).
- Physikalische oder logische Bindung des TPMs an die Plattform (z.B. mittels anlöten an das Motherboard oder Kryptographie).
- Plattform $\hat{=}$ Motherboard/IT-System.
- Plattform muss konform zur Evaluierungsrichtlinien der TCG sein
→ Conformance Credential (CC)

■ Platform Credential:

- Elektronisches Zertifikat vom Plattform-Hersteller.
- Bestätigt gültige Verbindung zwischen TPM und Plattform
→ Trusted Plattform.
- Bestandteile: Name des Plattformherstellers, Plattformmodell und Versionsnummer, Verweise auf die EC und CC.

→ **Schlüssel und deren Eigenschaften (1/7)**

- ❑ **Migratable Keys** → Auf andere Plattformen übertragbar (migrierbar).
- ❑ **Non-Migratable Keys** → An die Plattform gebunden (nicht migrierbar).

- ❑ **Storage Root Key (SRK):**
 - Wurzel der Schlüsselhierarchie.
 - Während der Installation des TPM-Eigentümers generiert.
 - Löschung des TPM-Eigentümers → Löschung des SRK
→ Kein Zugriff auf die Schlüsselhierarchie mehr.
 - **Eigenschaften:**
 - Steht im nicht flüchtigen Speicher des TPMs.
 - Ist nicht migrierbar.

→ **Schlüssel und deren Eigenschaften (2/7)**

□ **Attestation Identity Keys (AIK):**

- Verwendet für die Trusted Computing Funktion „**Attestation**“:
 - Authentische Bestätigung der Integrität einer Plattformkonfiguration (z.B. aktuelle Hard- und Softwareumgebung).
- Nötig, da EK datenschutzsensibel ist.
- AIKs werden vom TPM-Besitzer generiert.
- TPM/Plattform kann mehrere AIKs besitzen (z.B. für Online-Banking, E-Mail, ...)
- **Eigenschaften:**
 - Stehen im nicht flüchtigen Speicher des TPMs.
 - Nicht migrierbar.

→ Schlüssel und deren Eigenschaften (3/7)

□ Storage Keys (StorK):

- Verschlüsselung von weiteren Schlüsseln und Daten außerhalb des TPMs.
- Verwendet für die Trusted Computing Funktion „**Sealing**“:
 - Zustand der Plattform wird Teil der Verschlüsselung.
 - Entschlüsselung nur im vorher **definierten Zustand** (SW/HW) möglich.
- **Eigenschaften:**
 - RSA-Schlüsselpaar.
 - Im Allgemeinen darf die Migration zu anderen TPMs erfolgen.

→ **Schlüssel und deren Eigenschaften (4/7)**

□ **Binding Keys (BindK):**

- Verschlüsselung von beliebigen Daten außerhalb des TPMs.
- Entspricht der asymmetrischen Verschlüsselung.
- **Eigenschaften:**
 - RSA-Schlüsselpaar (es können auch andere Algorithmen vom TPM unterstützt werden).
 - Im Allgemeinen darf die Migration zu anderen TPMs erfolgen.
 - Kann nur mit Binding-Befehlen verwendet werden.

→ **Schlüssel und deren Eigenschaften (5/7)**

□ **Signing Keys (SigK):**

- Nachweis der Authentizität und Integrität von beliebigen Daten /Protokollnachrichten innerhalb und außerhalb des TPMs.
- **Eigenschaften:**
 - RSA-Schlüsselpaar (es können auch andere Algorithmen vom TPM unterstützt werden).
 - Im Allgemeinen darf die Migration zu anderen TPMs erfolgen.

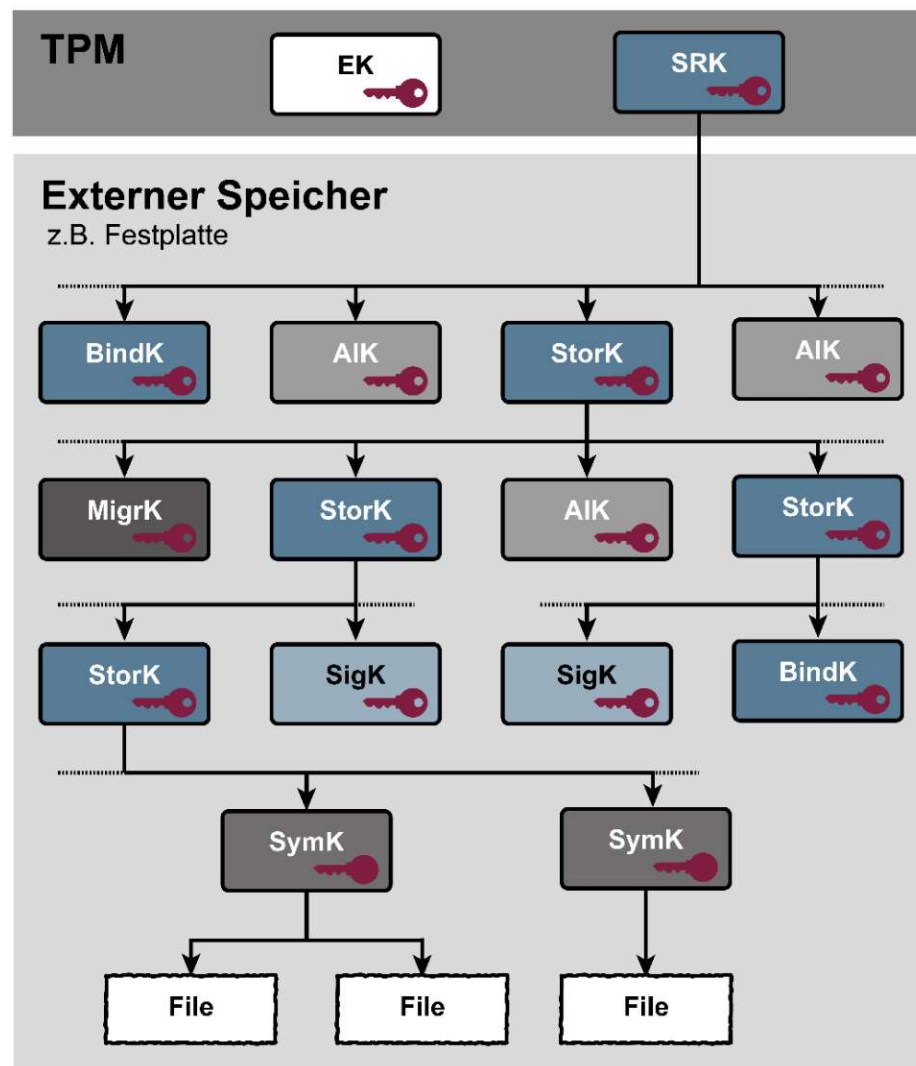
IT-Sicherheitsarchitektur

→ Schlüssel und deren Eigenschaften (6/7)




IT-Sicherheitsarchitektur

→ Schlüssel und deren Eigenschaften (6/7)



→ Schlüssel und deren Eigenschaften (7/7)

TPM Key Object 

General Information

Key Type

Algorithm

Authorization Secret

Specific Information

Key Length

Key Data

Key Properties

Migration

PCR Values

Trusted Computing

→ **Inhalt**

- ❑ Ziele und Ergebnisse der Vorlesung
- ❑ Einleitung
- ❑ IT-Sicherheitsarchitektur
- ❑ Trusted Computing-Funktion
- ❑ Trusted Network Connect - TNC
- ❑ Zusammenfassung

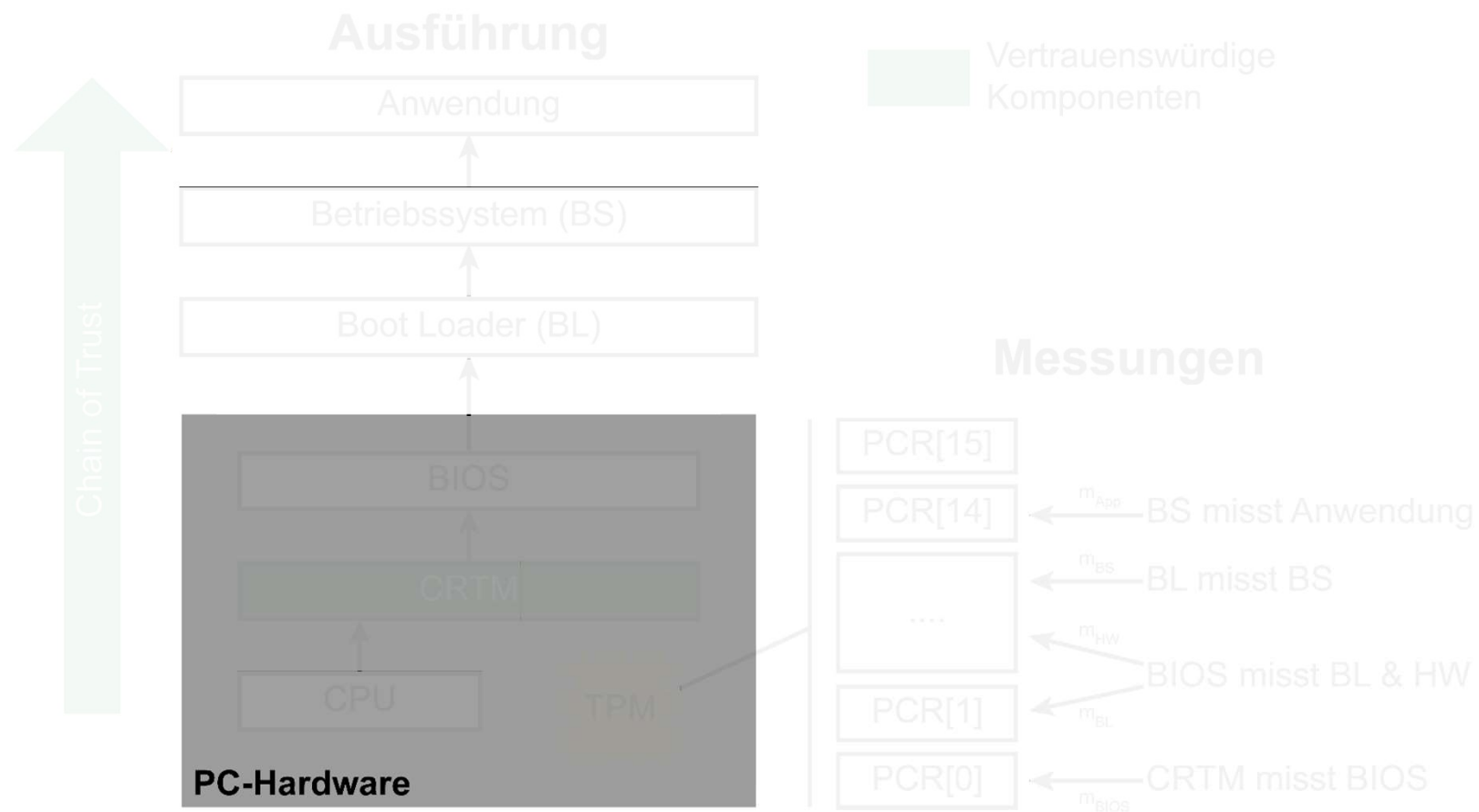
Trusted Computing-Funktion

→ **Authenticated Boot /Trusted Boot**

- Mithilfe von **Authenticated Boot** (oder Trusted Boot) kann dafür gesorgt werden, dass ein IT-System nur in einem **definierten vertrauenswürdigen Zustand** aktiv wird.
- Dabei werden beim Authenticated Boot die **Systemzustände der Soft- und Hardware** erst mal nur **gemessen** und in die **Platform Configuration Register (PCR) des TPMs gespeichert**.
- **Ablauf:**
Berechnung des Hashwertes und Schreiben in ein PCR-Register
 - Vertrauenswürdige CRTM Software → BIOS
 - BIOS → Boot Loaders, Hardwarezustand (Mainboard, ROM-Konfigu. ...)
 - Boot Loader → Betriebssystem
 - Betriebssystem → Anwendung
- Der **Gesamtwert** kann dann von einer **lokalen Instanz** wie ein Hardware-Sicherheitsmodul eines USB-Sticks oder **aus der Ferne** mit „Remote Attestation“ **überprüft werden**.

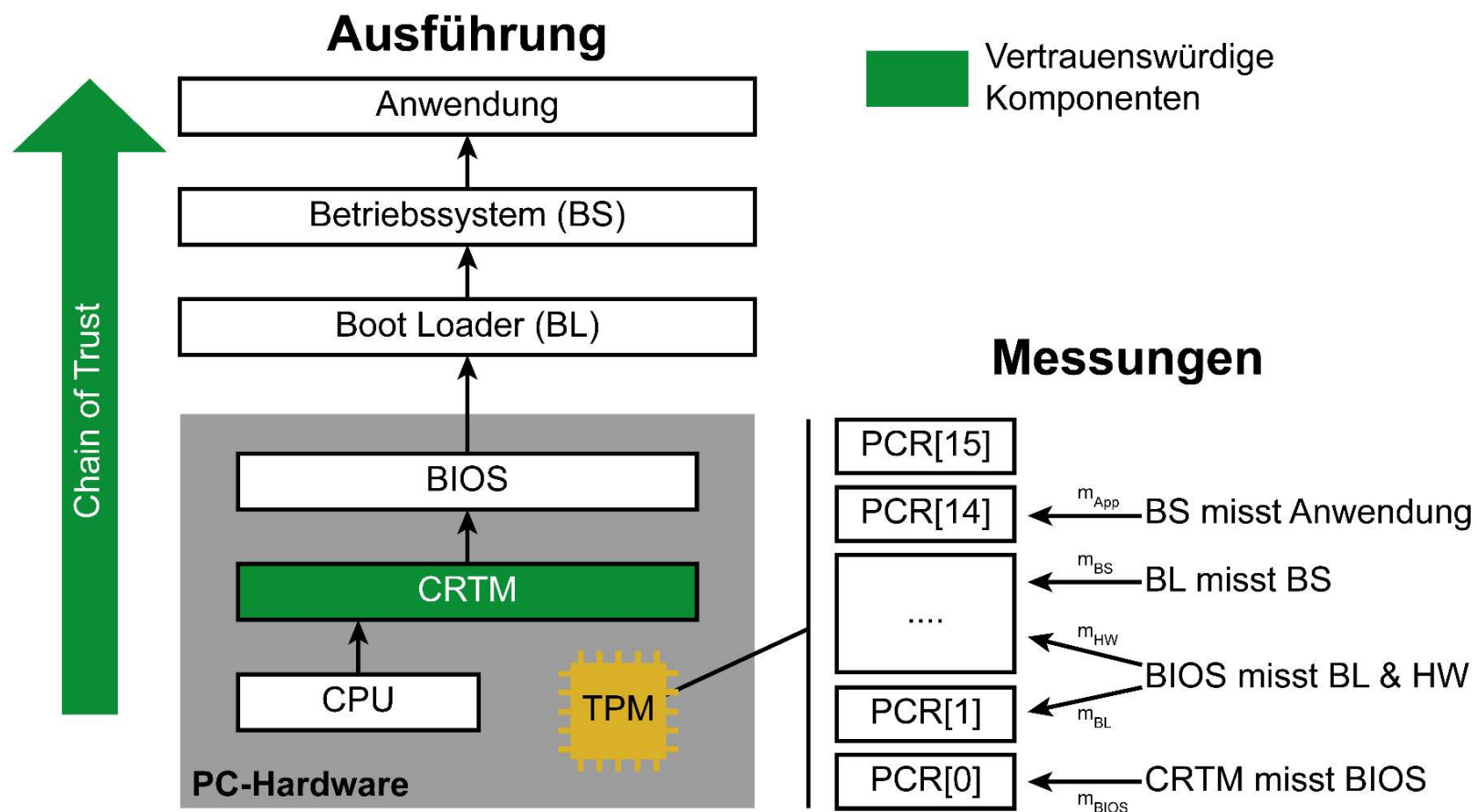
Sicherheitsarchitektur

→ **Authenticated Boot**



Sicherheitsarchitektur

→ Authenticated Boot



Trusted Computing-Funktion

→ **Binding**

- Binding wird verwendet, um **Daten** an eine bestimmte **TPM/Plattform** zu **binden**.
- Daten, die mit einem nicht migrierbaren Schlüssel verschlüsselt wurden, können nur von dem entsprechenden TPM wiederhergestellt werden, das den passenden geheimen Schlüssel kennt.
- **Normalerweise** bietet Binding **keine Plattformbindung**, da die Bindung auch mit **migrierbaren Schlüsseln** umgesetzt werden kann.
- Dadurch können **verschlüsselte Daten** auf eine **andere Plattform transferiert** und genutzt werden.

Trusted Computing-Funktion

→ Sealing

- **Sealing bindet Daten** immer an ein bestimmtes **TPM** und zusätzlich auch an die **Plattformkonfiguration** (Soft- und Hardwarekonfiguration).
- Sealing kann nur mit „**nicht migrierbaren Schlüsseln**“ umgesetzt werden.
- Damit kann die **Soft- und Hardwarekonfiguration** der Plattform verifiziert werden.
- Der **Schlüsseltext enthält implizit** auch den **Status der Plattform** (Soft- und Hardwarekonfiguration) zum Zeitpunkt der Verschlüsselung.
- Dadurch können mit Sealing **Daten an** eine bestimmte **Plattformkonfiguration** gebunden werden.
- Daten können nur entschlüsselt werden, wenn sich die Plattform (Soft- und Hardwarekonfiguration) in einem vordefinierten Zustand befindet.
- Die Idee ist, dass der vordefinierte Zustand ein **vertrauenswürdiger Zustand** ist, mit einer vorgegebenen vertrauenswürdigen Software und Hardware.

Trusted Computing-Funktion

→ Sealing Funktionen und Parameter

Eingabe Parameter

daten

{unverschlüsselte Daten}

Ausgabe Parameter

cipher

{verschlüsselte Daten}

crypteKEY

{verschlüsselter Schlüssel}

TPM Interne Funktionen und Daten

encrypt (key, daten)

{symmetrischer Verschlüsselungsalgorithmus „AES“}

H (daten)

{One-Way-Hashfunktion „SHA-256“}

genKey()

{Schlüsselerzeugung}

SRK

{Storage Root Key}

PCRs

{PCR-0, PCR-1, ...} z.B. aktuell abgespeicherte PCR-Werte

plainKEY = genKEY ()

cipher = encrypt (plainKEY, (daten //H (daten //PCR-0 //... //PCR-x))

crypteKEY = encrypt (SRK, plainKEY //H (plainKEY))

Trusted Computing-Funktion

→ Sealing Funktionen und Parameter

Eingabe Parameter

daten {unverschlüsselte Daten}

Ausgabe Parameter

cipher {verschlüsselte Daten}

crypteKEY {verschlüsselter Schlüssel}

TPM Interne Funktionen und Daten

encrypt (key, daten) {symmetrischer Verschlüsselungsalgorithmus „AES“}

H (daten) {One-Way-Hashfunktion „SHA-256“}

genKey() {Schlüsselerzeugung}

SRK {Storage Root Key}

PCRs {PCR-0, PCR-1, ...} z.B. aktuell abgespeicherte PCR-Werte

plainKEY = genKEY ()

cipher = encrypt (plainKEY, (daten //H (daten //PCR-0 //... //PCR-x))

crypteKEY = encrypt (SRK, plainKEY //H (plainKEY))

Trusted Computing-Funktion

→ Un-Sealing Funktionen und Parameter

Eingabe Parameter

<i>cipher</i>	<i>{verschlüsselte Daten}</i>
<i>crypteKEY</i>	<i>{verschlüsselter Schlüssel}</i>

Ausgabe Parameter

<i>daten</i>	<i>{unverschlüsselte Daten}</i>
--------------	---------------------------------

TPM Interne Funktionen und Daten

<i>decrypt (key, daten)</i>	<i>{symmetrischer Verschlüsselungsalgorithmus „AES“}</i>
<i>H (daten)</i>	<i>{One-Way-Hashfunktion „SHA-256“}</i>
<i>checkPCRs (Hash-Value)</i>	<i>{vergleicht PCRs-Inhalte mit Hash-Value}</i>
<i>SRK</i>	<i>{Storage Root Key}</i>
<i>PCRs</i>	<i>{PCR-0, PCR-1, ...}</i>

plainKEY = decrypt (SRK, crypteKEY)

daten //H (daten //PCR-0 //... //PCR-x) = decrypt (plainKEY, cipher)

if (checkPCRs (Hash-Value))

return daten

else

return ERROR

Trusted Computing-Funktion

→ Un-Sealing Funktionen und Parameter

Eingabe Parameter

<i>cipher</i>	<i>{verschlüsselte Daten}</i>
<i>crypteKEY</i>	<i>{verschlüsselter Schlüssel}</i>

Ausgabe Parameter

<i>daten</i>	<i>{unverschlüsselte Daten}</i>
--------------	---------------------------------

TPM Interne Funktionen und Daten

<i>decrypt (key, daten)</i>	<i>{symmetrischer Verschlüsselungsalgorithmus „AES“}</i>
<i>H (daten)</i>	<i>{One-Way-Hashfunktion „SHA-256“}</i>
<i>checkPCRs (Hash-Value)</i>	<i>{vergleicht PCRs-Inhalte mit Hash-Value}</i>
<i>SRK</i>	<i>{Storage Root Key}</i>
<i>PCRs</i>	<i>{PCR-0, PCR-1, ...}</i>

plainKEY = decrypt (SRK, crypteKEY)

daten //H (daten //PCR-0 //... //PCR-x) = decrypt (plainKEY, cipher)

```
if ( checkPCRs ( Hash-Value ) )  
    return daten  
else  
    return ERROR
```

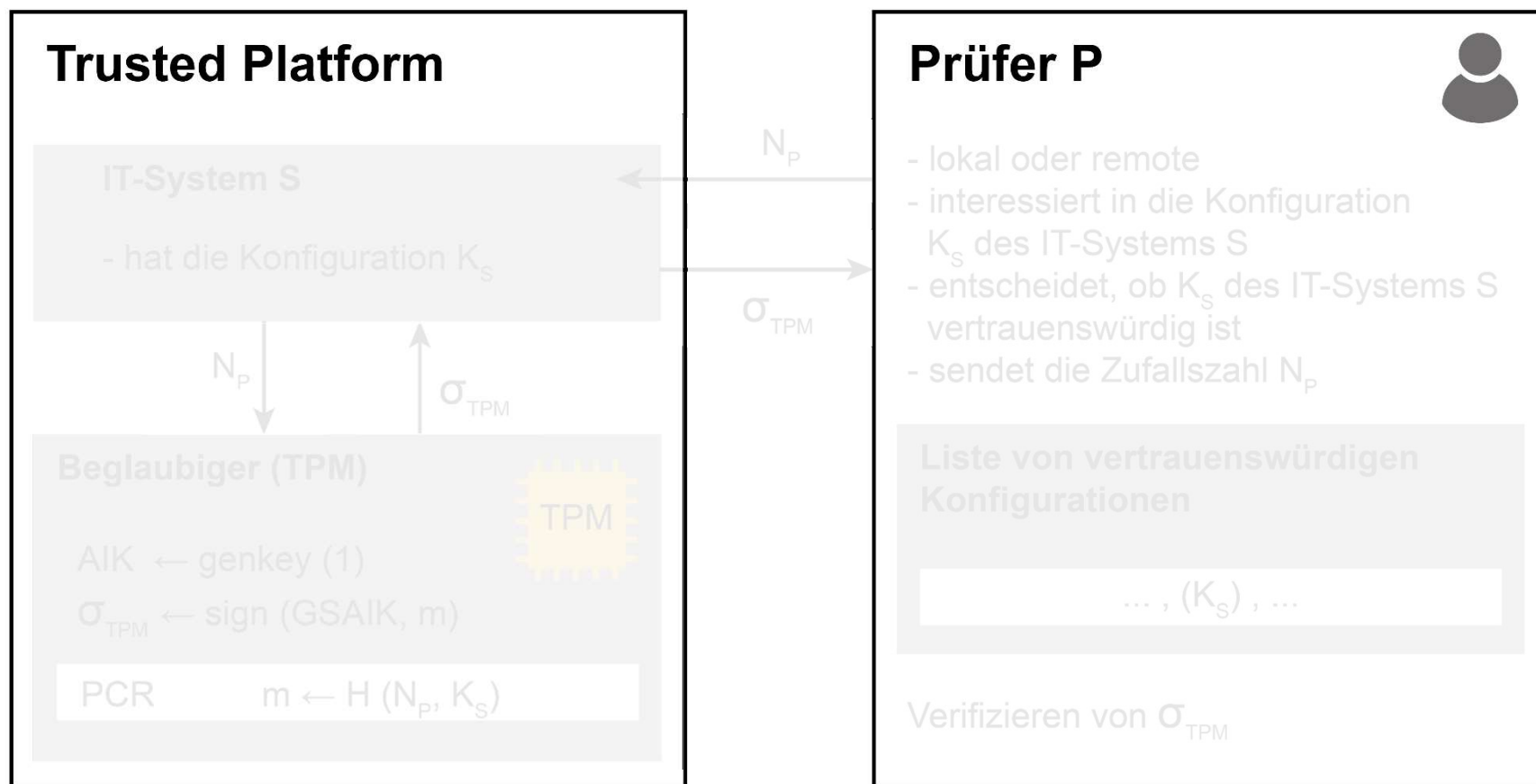
Trusted Computing-Funktion

→ **Remote Attestation**

- Mit (Remote) **Attestation** ist es möglich, die aktuelle Konfiguration eines IT-Systems zu überprüfen.
- Eine prüfende Instanz kann lokal oder remote überprüfen, ob eine gewünschte vertrauenswürdige Konfiguration eines IT-Systems vorliegt.
- Dazu hat die prüfende Instanz eine Liste von vertrauenswürdigen Konfigurationen, die ein IT-System haben kann, damit es gestartet werden kann oder eine Interaktion umgesetzt wird.
- **Remote Attestation** gibt die Möglichkeit, die **Vertrauenswürdigkeit** von **anderen, auch fremden IT-Systemen zu messen**, bevor eine Interaktion mit diesem IT-System begonnen wird.

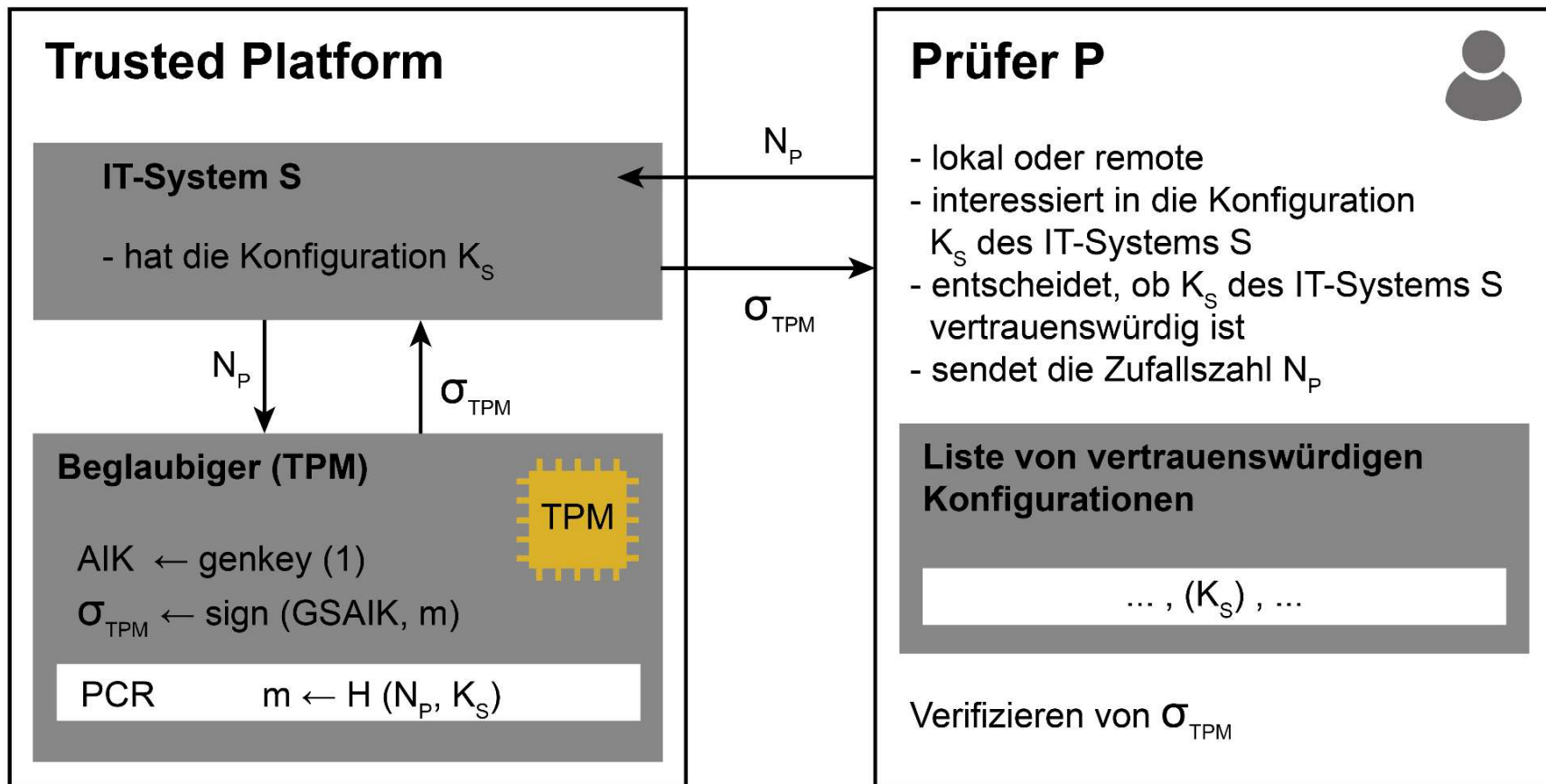
Trusted Computing-Funktion

→ (Remote) Attestation Ablauf



Trusted Computing-Funktion

→ (Remote) Attestation Ablauf



Trusted Computing-Funktion

→ Signaturfunktion für die Attestierung

Eingabe Parameter

random

{Zufallszahl des Prüfers $P - N_P$ }

Ausgabe Parameter

signature//certificate

{Signatur der aktuellen Systemkonfiguration des AIKs}

TPM Interne Funktionen und Daten

sign (key, daten)

{RSA-Signatur}

H (daten)

{One-Way-Hashfunktion "SHA-256"}

GSAIK

{geheimer AIK-RSA-Schlüssel}

AIK-certificate

{elektronisches Zertifikat des AIKs}

PCRs

{PCR-0, PCR-1, ...} z.B. aktuell abgespeicherte PCR-Werte

$$\sigma_{TPM} = \text{sign} (GSAIK, H (\text{random} // \text{PCR-0} // \dots // \text{PCR-x}))$$

Trusted Computing-Funktion

→ Signaturfunktion für die Attestierung

Eingabe Parameter

random

{Zufallszahl des Prüfers $P - N_P$ }

Ausgabe Parameter

signature//certificate

{Signatur der aktuellen Systemkonfiguration des AIKs}

TPM Interne Funktionen und Daten

sign (key, daten)

{RSA-Signatur}

H (daten)

{One-Way-Hashfunktion "SHA-256"}

GSAIK

{geheimer AIK-RSA-Schlüssel}

AIK-certificate

{elektronisches Zertifikat des AIKs}

PCRs

{PCR-0, PCR-1, ...} z.B. aktuell abgespeicherte PCR-Werte

$$\sigma_{TPM} = \text{sign} (GSAIK, H (\text{random} // \text{PCR-0} // \dots // \text{PCR-x}))$$

Trusted Computing-Funktion

→ Verifikationsfunktion für die Attestierung

Eingabe Parameter

signature//certificate

{Signatur der Systemkonfiguration des AIKs}

Ausgabe Parameter

return value

{Rückgabewert}

TPM Interne Funktionen und Daten

very (key, daten)

{RSA-Signatur-Verifikation}

H (daten)

{One-Way-Hashfunktion "SHA-256"}

ÖSAIK

{öffentlicher AIK-RSA-Schlüssel}

checkPCRs (PCR-Values)

{vergleicht den Inhalt der PCRs mit den gewünschten Werten}

PCRs

{PCR-0, PCR-1, ...}

if(very (ÖSAIK, σ_{TPM})) and

if(checkCERT (certificate)) and

if(checkPCRs (Hash-Value))

return ok

else

return ERROR

Trusted Computing-Funktion

→ Verifikationsfunktion für die Attestierung

Eingabe Parameter

signature//certificate

{Signatur der Systemkonfiguration des AIKs}

Ausgabe Parameter

return value

{Rückgabewert}

TPM Interne Funktionen und Daten

very (key, daten)

{RSA-Signatur-Verifikation}

H (daten)

{One-Way-Hashfunktion "SHA-256"}

ÖSAIK

{öffentlicher AIK-RSA-Schlüssel}

checkPCRs (PCR-Values)

{vergleicht den Inhalt der PCRs mit den gewünschten Werten}

PCRs

{PCR-0, PCR-1, ...}

if (very (ÖSAIK, σ_{TPM})) and

if (checkCERT (certificate)) and

if (checkPCRs (Hash-Value))

return ok

else

return ERROR

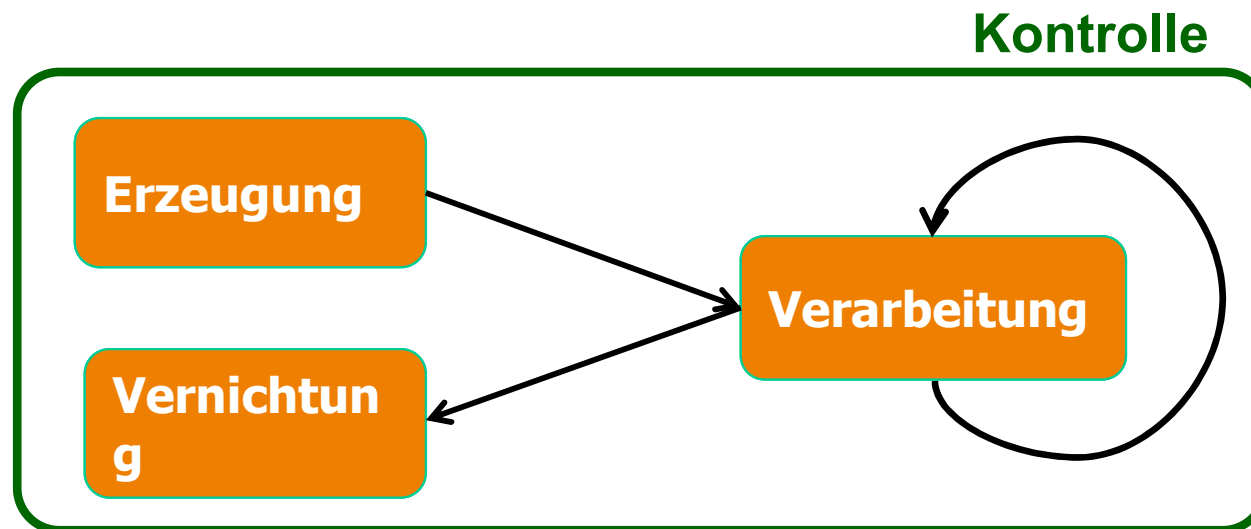
Objekt-Sicherheit

→ Remote Attestation

Idee: Domänenorientierte Objektsicherheit, bei der die Objekte mit Rechten versehen werden, die definieren, wer sie in welcher IT-Umgebung wie nutzen darf.

→ *Object Lifecycle Protection*

→ *Distributed Policy Enforcement (**even on foreign systems**)*



Festlegung einer sicheren

→ **Systemkonfiguration (1/2)**

- ❑ Mit Trusted Computing können Hard- und Software-Systemkonfigurationen gemessen und jederzeit überprüft werden.
- ❑ Eine wichtige Frage ist aber, wer definiert, was eine sichere und vertrauenswürdige Systemkonfiguration ist.
- ❑ **Der Hersteller der IT-Systems?**
 - Der Hersteller **kennt die Hardware sehr gut**, weil er sie selber baut oder bauen lässt.
 - Er ist für das **Betriebssystemen** und **Basis-Anwendungen** verantwortlich.
 - Mit diesem Wissen und Gestaltungsspielraum kann jeder **Hersteller** genau **festlegen, was eine sichere und vertrauenswürdige Systemkonfiguration ist.**

Festlegung einer sicheren

→ **Systemkonfiguration (2/2)**

❑ **Der Hersteller des Cyber-Sicherheitssystems?**

- Er kennt sich mit den **Cyber-Sicherheitsproblemen** und **Angriffsvektoren** aus und kann auf dieser Basis definieren, was eine sichere und vertrauenswürdige Systemkonfiguration ist.

❑ **Das Anwendungsunternehmen der IT-Systeme?**

- Das Anwendungsunternehmen hat **Erfahrungen mit der Nutzung** der IT-Systeme, kennt die **realen Angriffe** und die daraus **verursachten Schäden**.
- Mit diesen praktischen Erfahrungen kann das Anwendungsunternehmen sichere und vertrauenswürdige Systemkonfigurationen definieren.

❑ **Kooperation der unterschiedlichen Rollen**

- Ideal wäre es, wenn sich die **Hersteller** der IT-Systeme und Cyber-Sicherheitssysteme sowie die **Anwendungsunternehmen** zusammentun würden, um eine sichere und vertrauenswürdige Systemkonfigurationen der genutzten IT-Systeme zu definieren.

Trusted Computing

→ **Inhalt**

- ❑ Ziele und Ergebnisse der Vorlesung
- ❑ Einleitung
- ❑ IT-Sicherheitsarchitektur
- ❑ Trusted Computing-Funktion
- ❑ Trusted Network Connect - TNC
- ❑ Zusammenfassung

Trusted Computing

→ **Inhalt**

- ❑ Ziele und Ergebnisse der Vorlesung
- ❑ Einleitung
- ❑ IT-Sicherheitsarchitektur
- ❑ Trusted Computing-Funktion
- ❑ Trusted Network Connect - TNC
- ❑ Zusammenfassung

Trusted Computing

→ Zusammenfassung (1/2)

- ❑ Die Kernfunktionalitäten von Trusted Computing sind:
 - **Robustheit und Modularität**
 - **Integritätsüberprüfung**
 - **Trusted Process**
 - **Trusted Plattform**
- ❑ Vor- und Nachteile der verschiedenen **Kernelarchitekturen** müssen miteinander abgewogen werden.
- ❑ **CRTM** ist die Vertrauensbasis. Das Vertrauen ist **transitiv**.
- ❑ Die **TPM Schlüsselhierarchie** ermöglicht eine sichere Speicherung von Daten, auch auf externen Speichermedien.

Trusted Computing

→ Zusammenfassung (2/2)

- ❑ Wichtige Trusted Computing Funktionen sind:
 - **Authenticated Boot**
 - **Binding**
 - **Sealing**
 - (remote) **Attestation**
- ❑ Vertrauenswürdige Netzwerkverbindungen können durch **Trusted Network Connect (TNC)** realisiert werden.
- ❑ Die Festlegung einer sicheren und vertrauenswürdigen **Systemkonfiguration** ist mit zahlreichen Schwierigkeiten (technisch und politisch) verbunden.

Netzwerksicherheit

Anhang / Credits / Quellen

- **Cyber-Sicherheit**
Das **Lehrbuch** für Konzepte, Mechanismen, Architekturen und Eigenschaften von Cyber-Sicherheitssystemen in der Digitalisierung“, Springer Vieweg Verlag, Wiesbaden 2022
<https://norbert-pohlmann.com/cyber-sicherheit/>



Vielen Dank fürs Ansehen!

□ Fragen?

□ Chat, Tutorium, Forum – Sie haben die Wahl!

Questions?