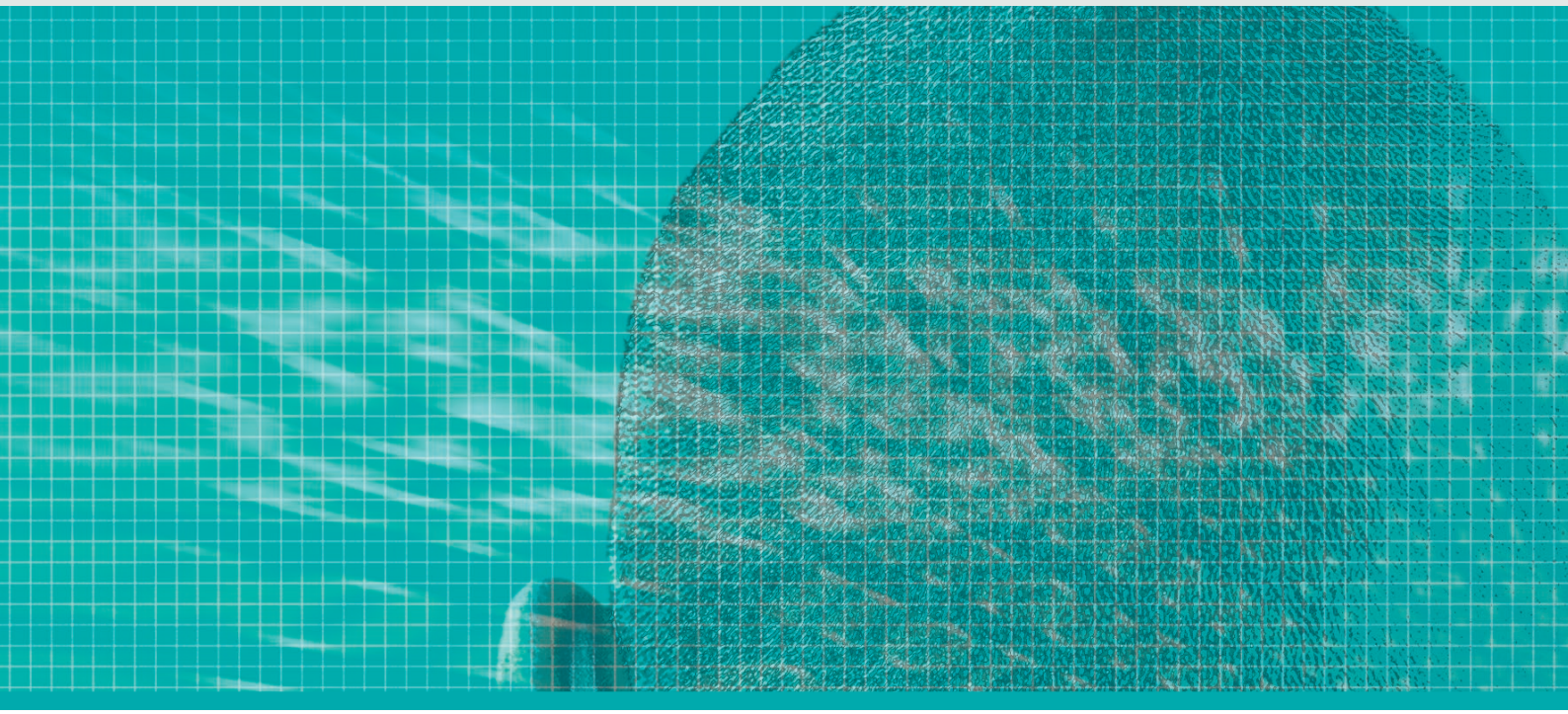
White Paper



Software-Integritätsschutz

Das Leitbild von Wibu-Systems für die Cyber-Allianz

Oliver Winzenried, Vorstand WIBU-SYSTEMS AG
www.wibu.com

WIBU SYSTEMS

Inhalt

Die CyberAllianz für Cyber-Sicherheit	3
Integritätsschutz für Embedded-Systeme	4
Was ist Integritätsschutz?	4
Wo sind Berührungspunkte zum Kopier- und Know-how-Schutz?	4
Grundbegriffe der Kryptographie	5
Angriffe auf Cyber-Physical Systems	6
Aufbau eines Embedded-Systems und dessen Herausforderungen	6
Realisierung der Integritätsprüfung	7
Rückwärtsprüfung (Backward Check)	8
Pre-Bootloader – Der erste Schritt	9
Zertifikatskette	9
Zusammenfassung	11

Autor:
Oliver Winzenried ist begeisterungsfähiger Verfechter von Sicherheitslösungen, die gepaart mit innovativen Technologien das geistige Eigentum und Umsätze unabhängiger Software-Hersteller schützen. Unmittelbar nach Abschluss seines Elektrotechnikstudiums an der Universität Karlsruhe begann er seine unternehmerische Laufbahn in der Entwicklung von Elektronik- und ASIC-Bausteinen, Hardware, Mikrocontroller und Eingebetteter Systeme für die Bereiche Unterhaltungselektronik, Automobil- und Betriebstechnik. In 1989 gründete er zusammen mit Marcellus Buchheit die WIBU-SYSTEMS AG, deren Geschäftsführer er seitdem ist. Seine Leidenschaft für den Integritätsschutz von Software findet ihren Ausdruck in zahlreichen Patenten, die vom sicheren Lizenzmanagement bis zu Produktinnovationen bei Dongles reichen. Als Autor liefert er regelmäßig Beiträge zu Leitartikeln und Büchern und seine Vorträge finden die Aufmerksamkeit auf großen Messen, Ausstellungen, Konferenzen, Industrieverbandsveranstaltungen und Technologiezentren wie das Fraunhofer Institut. Sein persönliches Engagement in internationalen Projekten im F&E-Bereich und Standardisierungsgremien, wie z.B. die SD Card Association, runden sein Profil ab. Oliver Winzenried ist überdies Vorstandsvorsitzender der Arbeitsgemeinschaft Produkt- und Know-how-Schutz „Protect-Ing“ des VDMA, im Hauptvorstand der BITKOM sowie im Vorstand des Fördervereins Forschungszentrum Informatik FZI am KIT.

Die CyberAllianz für Cyber-Sicherheit

WIBU-SYSTEMS engagiert sich aktiv in der CyberAllianz für Cyber-Sicherheit, einer Initiative des Bundesamtes für Sicherheit in der Informationstechnik (BSI), die in Zusammenarbeit mit dem Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V. (BITKOM) gegründet wurde.

Die Allianz hat das Ziel, aktuelle und valide Informationen flächendeckend bereitzustellen. Sie baut auf der Basis eines großen Wissensschatzes und der Erfahrung der Mitglieder eine umfangreiche Wissensbasis für Teilnehmer auf und unterstützt dadurch wesentlich den Informations- und Erfahrungsaustausch. Das folgende Dokument leistet dabei einen grundlegenden Betrag für das wichtige Verständnis des Einsatzes von Integritätsschutz bei elektronischen Rechnern im rein technischen Umfeld, den eingebetteten (engl. embedded) Systemen. Auf der einen Seite wird dadurch das mit diesen Steuerungen ausgelieferte Know-how vor Reverse-Engineering geschützt, auf der anderen Seite begegnet man dadurch den immer größer werdenden Bedrohungen durch Wirtschaftskriminalität und des Cyber-Wars.

WIBU SYSTEMS

Allianz für Cyber-Sicherheit Partner

Integritätsschutz für Embedded-Systeme

Steuerungssysteme sind zunehmend vernetzt und kommunizieren über öffentliche Netze. Mit Cyber-Physical Systems fusionieren die analoge Welt mit der physischen und die digitale Welt mit der virtuellen Realität. Dieser auch als Industrie 4.0 bezeichnete Trend bietet aber nicht nur Komfort und Mehrwert, sondern öffnet die Systeme nach außen und erhöht damit die Gefahr von Angriffen von außen. Durch Integritätsschutz-Maßnahmen wird die Verfügbarkeit und Sicherheit, und zwar Safety und Security, der Systeme nachhaltig gewährleistet.

Was ist Integritätsschutz?

Unter dem Begriff Integritätsschutz versteht man Sicherheitsmaßnahmen, die Systemressourcen und Programme sowie Daten gegen unberechtigte Manipulation schützen oder deren Veränderung erkennen und anzeigen. Die Herausforderung besteht darin, die Integrität der Daten zu gewährleisten und falls nicht, das System in einen sicheren Zustand zu bringen und keine anderen Funktionen mehr auszuführen. Die Lösungen basieren auf Kryptografie und damit einhergehenden Sicherheits-Mechanismen, wie z.B. digitalen Signaturen und Message-Authentification.

Wo sind Berührungspunkte zum Kopier- und Know-how-Schutz?

Unter Kopierschutz versteht man den Produktschutz gegen Nachbau kompletter Maschinen und Geräte und unter Know-how-Schutz den Schutz der Algorithmen und Verfahren gegen Reverse-Engineering.

Um der zunehmenden Produktpiraterie entgegenwirken zu können, gewinnt auch der Schutz von Software und Daten dieser Systeme immer mehr an Bedeutung. Software stellt bereits heute einen erheblichen Wirtschaftsfaktor dar und der Software-Anteil an Innovationen im Maschinen- und Anlagenbau nimmt stetig zu. Im Automotive-Bereich sind heute 90% aller Innovationen von Elektronik und Software getrieben und der auf Software basierende Wertschöpfungsanteil wird in den nächsten Jahren auf 40% steigen.

Die Ergebnisse der Umfrage des VDMA aus dem Jahr 2012 weisen einen Umsatzausfall in Höhe von 7,9 Mrd. € im Jahr 2012 aus, 9 von 10 Unternehmen sind von Produktpiraterie betroffen und 48% der Hersteller leiden unter dem Nachbau kompletter Maschinen. 28% der Befragten gaben an, technische Schutzlösungen einsetzen zu wollen. 37.000 Arbeitsplätze könnten im deutschen Maschinen- und

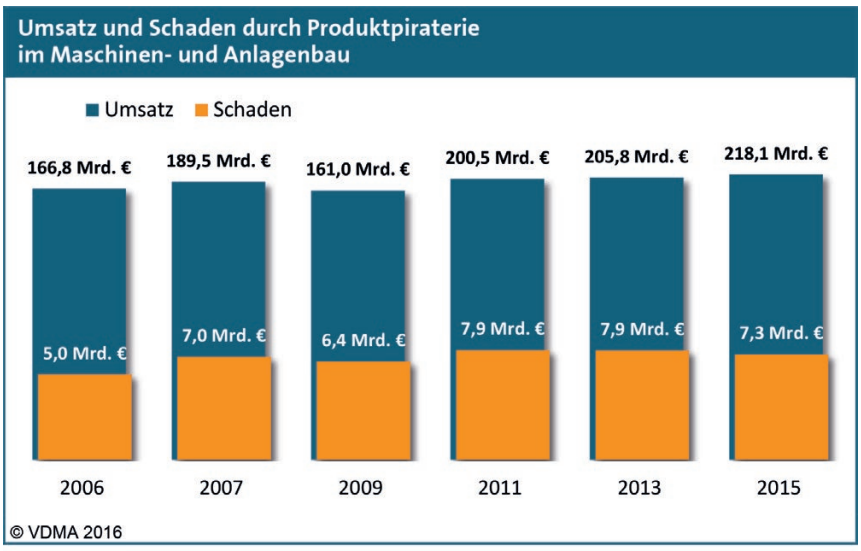


Abbildung 1: Umsatz und Schaden durch Produktpiraterie (Quelle: VDMA)

Branchenumsatz und Schaden durch Produktpiraterie in Deutschland im Vergleich

N=193

Anlagenbau zusätzlich geschaffen werden, wenn es gelänge, die Produktpiraterie zu reduzieren. Die Abbildung 1 zeigt den ansteigenden Trend in den letzten Jahren. Ein wirkungsvoller Schutz der Software ist folglich die Voraussetzung für den Schutz der Produkt-Innovationen. Gleichzeitig steigt mit zunehmender Digitalisierung der Produktion auch die Bedeutung des Schutzes von digitalen Produktionsdaten und ganz allgemein mit zunehmender Vernetzung der Schutz der Daten und der Integrität der Systeme. Neue, auf starker Kryptografie unter Verwendung sicherer Hardware-Elemente (Secure Elements, Smart Card Chips) aufbauende Lösungen können gleichermaßen zum Kopier- und Know-how-Schutz und zum Integritätsschutz eingesetzt werden.

Grundbegriffe der Kryptografie

Zum besseren Verständnis der nachfolgend beschriebenen Verfahren hier eine kurze Erläuterung einiger Begriffe:

Symmetrische Kryptografie: Beim symmetrischen Kryptosystem verwenden beide Teilnehmer den gleichen Schlüssel oder einen einfach ableitbaren Schlüssel. Beispiele für symmetrische Kryptosysteme sind FEAL (Fast Encryption Algorithm), IDEA (International Date Encryption Algorithm), DES (Data Encryption Standard) oder AES (Advanced Encryption Standard). Der Vorteil symmetrischer Kryptosysteme ist eine hohe Verschlüsselungsgeschwindigkeit, der Nachteil oder die Herausforderung besteht im sicheren Schlüsselaustausch.

Asymmetrische Kryptografie: Ein asymmetrisches Kryptosystem oder Public-Key-Kryptosystem ist ein kryptografisches Verfahren, bei dem im Gegensatz zu einem symmetrischen Kryptosystem die kommunizierenden Parteien keinen gemeinsamen geheimen Schlüssel kennen müssen. Ein Benutzer erzeugt hier ein Schlüsselpaar, das aus einem geheimen Teil, dem privaten Schlüssel, und einem nicht geheimen Teil, dem öffentlichen Schlüssel, besteht. Der öffentliche Schlüssel ermöglicht es jedem, Daten für den Inhaber des privaten Schlüssels zu verschlüsseln, dessen digitale Signaturen zu prüfen oder ihn zu authentifizieren. Der private Schlüssel ermöglicht es seinem Inhaber, mit dem öffentlichen Schlüssel verschlüsselte Daten zu entschlüsseln, digitale Signaturen zu erzeugen oder sich zu authentisieren. Beispiele für asymmetrische Algorithmen sind RSA (Rivest, Shamir und Adleman) oder ECC (Elliptic Curve Cryptography).

Hash-Funktionen: Eine Hash-Funktion ist eine Funktion, die eine Zeichenfolge beliebiger Länge auf eine Zeichenfolge mit fester Länge abbildet. Eine kryptografische Hash-Funktion ist eine spezielle Form der Hash-Funktion, welche zusätzlich kollisionsresistent oder eine Einwegfunktion (oder beides) ist. Angewandt werden Hash-Funktionen zur Integritätsprüfung von Dateien oder Nachrichten, zur Verschleierung von Passwortdateien, als Datenbasis digitaler Signaturen, als Pseudo-Zufallszahlengeneratoren oder zur Konstruktion von Blockchiffren. Beispiele sind MD5 sowie SHA-1 und SHA-256.

Digitale Zertifikate: Ein Digitales Zertifikat ist ein digitaler Datensatz, der bestimmte Eigenschaften von Personen oder Objekten bestätigt und dessen Authentizität und Integrität durch kryptografische Verfahren geprüft werden kann. Das digitale Zertifikat enthält insbesondere die zu seiner Prüfung erforderlichen Daten. Weit verbreitet sind Public-Key-Zertifikate nach dem Standard X.509, welche die Identität des Inhabers und weitere Eigenschaften eines öffentlichen kryptografischen Schlüssels bestätigen.

Angriffe auf Cyber-Physical Systems

Um effektive Methoden zur Abwehr von Angriffen entwickeln zu können, sollten die Bedrohungsszenarien bekannt sein. Nachfolgend sind einige der möglichen Angriffe auf Embedded-Systeme aufgelistet:

1) Angreifer entwickeln ein „**Fake-Device**“, also ein Gerät, das genauso aussieht wie das Original, dessen Funktion aber manipuliert ist und das beispielsweise im Servicefall als Austauschteil eingebaut wird.
2) Angreifer entwickeln eine **eigene Software** und bringen diese durch Tausch der Speicherkarte in das Embedded-System.
3) Angreifer entnehmen die Speicherkarte aus dem Embedded-System, **manipulieren** die Software darauf und setzen sie wieder ins System ein.
4) Angreifer ändern die Software auf dem Embedded-System durch Angriffe über die **Kommunikationsschnittstellen** von außen.
5) Angreifer kommen in den Besitz eines **Embedded-Systems**, so wie es in der Anwendung verwendet wird, um es zu analysieren und Angriffswege zu entwickeln.

Aufbau eines Embedded-Systems und dessen Herausforderungen

Prinzipiell ist ein typisches Embedded-System in verschiedene Schalen aufgebaut. Dabei ist zu berücksichtigen, dass eine äußere Schale auf den Speicher einer inneren Schale zugreifen darf, umgekehrt ist das in den meisten Fällen nicht möglich. Die äußere Schale (Hardware/Bootlader) ist dabei die initiale Schale des Gesamtprozesses. Folgender Ablauf ergibt sich für einen zu realisierenden Integritätsschutz.

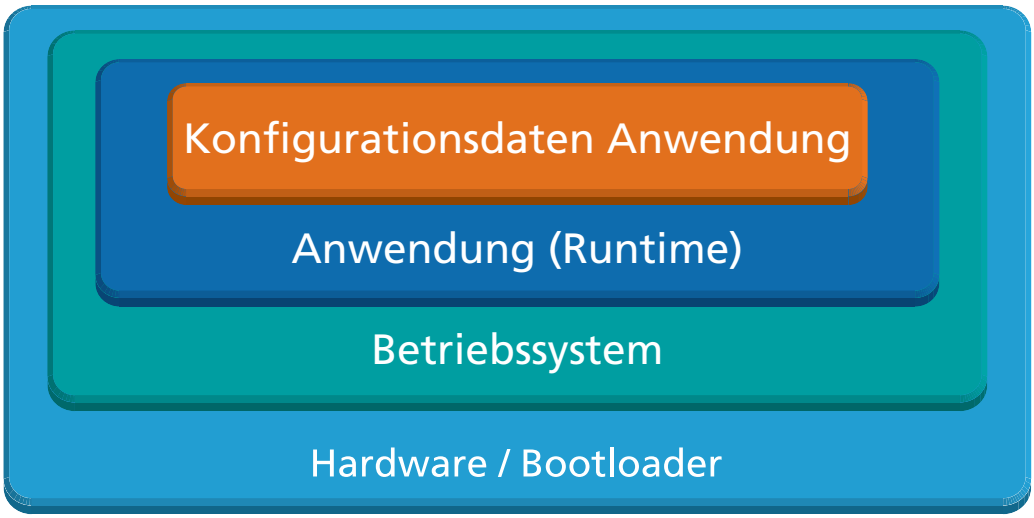


Abbildung 2: Prinzipieller Aufbau eines Embedded-Systems

Die einzelnen Schritte zeigen jeweils, wie die Integrität von Stufe zu Stufe des Systems sichergestellt werden kann. Wie die einzelnen Schritte realisiert werden, wird in den darauffolgenden Abschnitten näher beschrieben:

1) Der **Bootloader** überprüft die Integrität des Betriebssystems und lädt es, wenn es als korrekt eingestuft wurde.
2) Das **Betriebssystem** startet nur, wenn durch eine Rückwärtsprüfung der Bootloader als vertrauenswürdig bestätigt wurde.
3) Das Betriebssystem prüft die Integrität der Anwendung und lädt diese nur, wenn sie als korrekt angesehen wurde.
4) Die **Anwendung** startet nur, wenn durch eine Rückwärtsprüfung das Betriebssystem als vertrauenswürdig bestätigt wurde.
5) Die **Anwendung** überprüft die Integrität der Konfigurationsdaten und verwendet diese nur, wenn sie korrekt sind.
6) Sollten die **Konfigurationsdaten** auch ausführbaren Code enthalten, so ist auch eine Überprüfung der Vertrauenswürdigkeit der Anwendung möglich.

Realisierung der Integritätsprüfung

Zunächst muss die ungeschützte Original-Software gemäß dem nachfolgenden Ablauf signiert und verschlüsselt werden:

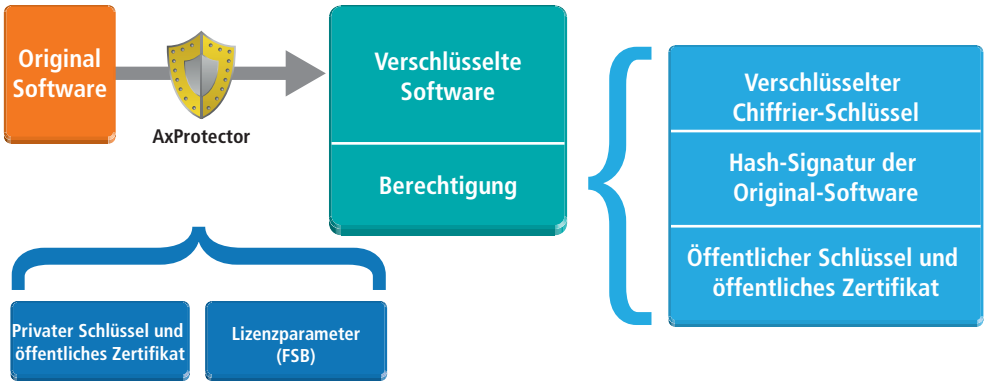


Abbildung 3: Verschlüsseln und Signieren einer Anwendung

Der AxProtector, ein kommerzielles Tool zum Schützen von Software, wird dabei für folgende Schritte verwendet:

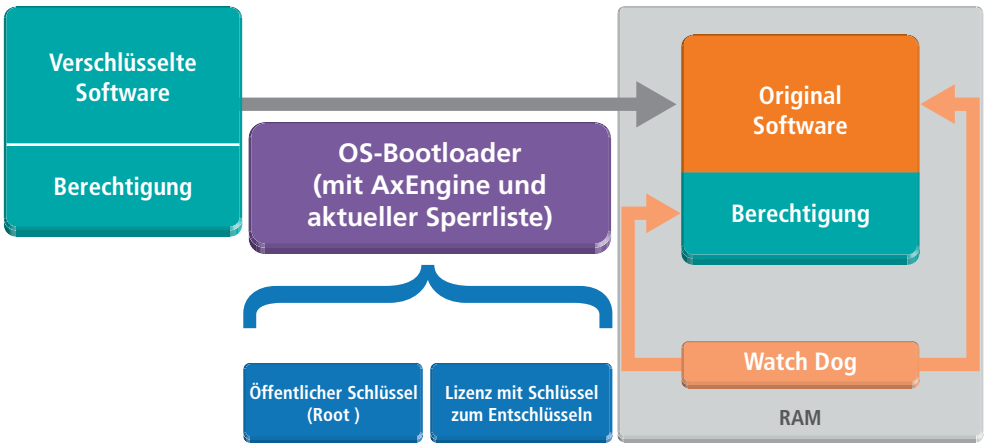


Abbildung 4: Integritätsprüfung (Forward-Check)

1) Berechnung des **Hashwerts** der Original-Software.
2) **Signieren** des Hashwerts mit dem privaten Schlüssel (Private Key) des Herausgebers.
3) **Verschlüsselung** der Original-Software unter Verwendung eines Schlüssels, der aus einem Seedwert aus der Original-Software, einem geheimen Schlüssel des Herausgebers und einigen Parametern, die der Herausgeber wählt, erzeugt wird.
4) Anhängen des öffentlichen Teils des **Signaturzertifikats** an die verschlüsselte Software.

Der erste Teil der Integritätsprüfung nach vorn (Forward Check), d.h. die Prüfung der zu ladenden Software oder der zugehörigen Daten, wird wie folgt durchgeführt:

Die Prüfung besteht aus den nachfolgenden Schritten, die beim Laden der Anwendung ausgeführt werden. Dies wird auf Basis eines in das System integrierten Tools durchgeführt, der AxEngine von Wibu-Systems.

1) Beim Vorhandensein einer passenden Lizenz wird die verschlüsselte Software **entschlüsselt**.
2) Das in den Credentials angehängte Zertifikat oder die Zertifikatskette wird gegen den öffentlichen Ursprungsschlüssel (Public Root Key) **geprüft**.
3) Der Hash-Wert über die entschlüsselte Original-Software wird **berechnet**.
4) Die Signatur über den Hash wird mit dem öffentlichen Schlüssel (Public Key) **überprüft**.

Zusätzlich zu diesen notwendigen Schritten können zur weiteren Erhöhung der Sicherheit noch weitere Maßnahmen umgesetzt werden, z.B. eine ausgefeilte Handhabung der Zertifikate auf Zulässigkeit in bestimmten Geräten. Ebenso kann auf ein spezifisches Ablaufdatum des Zertifikats oder auf das Vorhandensein des Zertifikats auf einer Sperrliste (Revocation list) geprüft werden. Dazu sind auch periodische Überprüfungen zur Laufzeit im Arbeitsspeicher des Systems denkbar (Watchdog).

Die Lösung mit der CodeMeter-Technologie führt folgende Schritte durch, die insbesondere auch in automatisierten Build-Prozessen ablaufen können:

- **Verschlüsselung** des Programmcodes, um statische Codeanalyse und ReverseEngineering zu verhindern.
- **Signieren** des Programmcodes sowohl von Anwendungen als auch vom Betriebssystem-Image.
- Speichern der geteilten **Geheimnisse** für die Entschlüsselung.
- Speichern der privaten **Signaturschlüssel** auf Erstellerseite.
- **Überprüfung** der Signaturen und Hash-Werte beim Laden und zur Laufzeit.

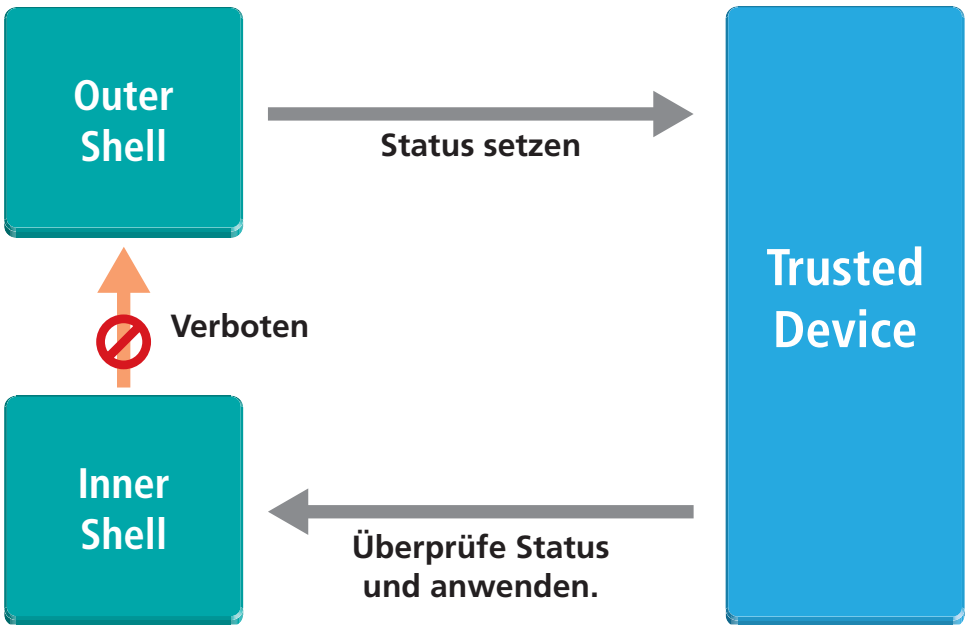
Rückwärtsprüfung (Backward Check)

Die Prüfung, ob der Boot-Prozess durch das Betriebssystem korrekt durchgeführt wurde oder die Integritätsprüfung des Betriebssystems durch die Anwendung, ist nur schwierig durchführbar, da der nachfolgende Schritt jeweils nur begrenzten Zugriff auf den vorherigen Schritt hat.

Um diesen mangelnden Zugriff zu kompensieren, wird eine Zustandsmaschine in einer vertrauenswürdigen Hardware benötigt. Konzepte dafür findet man bei der Trusted Computing Group, TCG.

Mittels sogenannter Trusted Plattform Modules (TPM) ist es möglich, korrekte Zustände in Registern zu speichern. Diese Register enthalten beispielsweise Messwerte des Bootloaders, die dann später vom Betriebssystem geprüft werden, um die Integrität des vorhergehenden Schrittes zu prüfen.
Die nachfolgende Grafik zeigt den Ablauf einer Rückwärtsprüfung mittels eines „Trusted Devices“, das den aktuellen Status verwaltet:

Abbildung 5: Rückwärtsprüfung über ein Trusted Device



Die „Inner Shell“ ist dabei beispielsweise das Betriebssystem, die „Outer Shell“ der Bootloader. Das Trusted Device, z.B. ein TPM Chip oder ein CodeMeter Dongle, speichert den Zustand des Bootlaoders. Nur wenn dieser korrekt durchlaufen wird, kann das Betriebssystem anschließend starten. Dies gilt für die nachfolgenden Stufen entsprechend.

CodeMeter bietet hierfür ebenso eine sichere Zustandsmaschine. Das Feature wird „Enabling“ genannt. Eine Entschlüsselung des Betriebssystems wird beispielsweise erst freigegeben, wenn der Bootprozess mit korrekter Integrität durchgeführt wurde, außerdem werden gemeinsame Geheimnisse gespeichert und erst bei entsprechend erfolgreich erfolgtem vorherigem Schritt für den nächsten Schritt freigegeben.

Pre-Bootloader – Der erste Schritt

Ein sicherer erster Schritt ist unbedingt erforderlich, da genau darauf die Überprüfung der Folgeschritte beruht. Angreifer dürfen in keinem Fall in der Lage sein, den Code zu entschlüsseln oder geheime Schlüssel zu extrahieren. Eine Lösung dafür sind sogenannte „System on Chip“ (SOC), bei denen, nicht von außen lesbar und nicht änderbar, dieser Code und Schlüssel fest auf dem Chip eingebrannt sind.

Dieser kleine Pre-Bootloader hat nur geringe Funktionalität und lädt den eigentlichen Bootloader, dessen Integrität er sicherstellen kann. Er wird nur einmal entwickelt und ist auf dem System nicht updatebar, um Angriffe von außen darauf zu verhindern.

Zertifikatskette

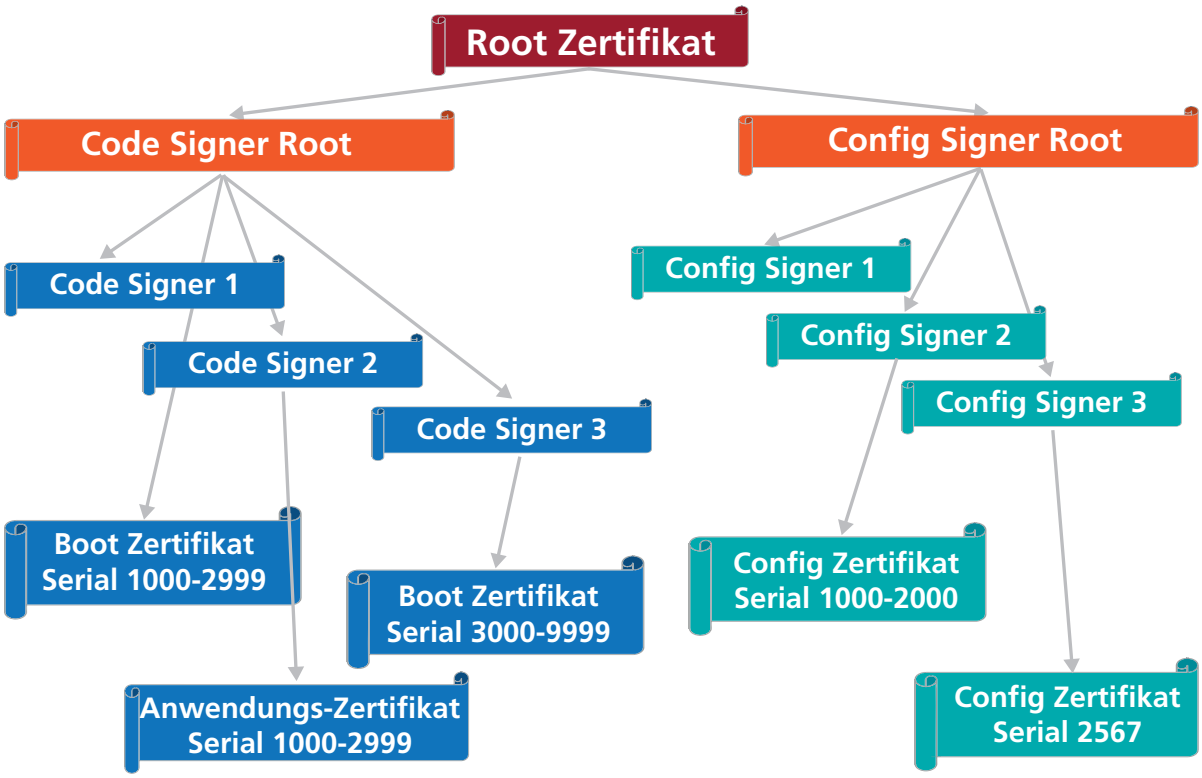
Für das Signieren des Programmcodes und auch von Parametern werden private Schlüssel, Private Keys, verwendet. Diese werden in einer sicheren Hardware, beispielsweise in einem CodeMeter Dongle, gespeichert. Die Zertifikate werden als Dateien gespeichert. Sie enthalten:

- Den **Public Key**, öffentlicher Schlüssel.
- Einschränkungen der **Gültigkeit**, z.B. Ablaufdatum oder Bindung an bestimmtes Gerät.
- **Verwendungszweck**, beispielsweise zur Signatur des Bootloader, Signatur der Anwendung, der Konfigurationsdateien oder zum Erstellen weiterer Zertifikate.
- **Zertifikat** des Schlüssels, das zum Erstellen dieses Zertifikats verwendet wurde.

Dies erscheint auf den ersten Blick kompliziert, macht aber Sinn. In der Praxis wird das sogenannte Root-Zertifikat, das ganz oben in der Kette steht, nur verwendet, um die später verwendeten Zertifikate zu erstellen. Das Root-Zertifikat wird sicher verwahrt und wird ausschließlich dann benötigt, wenn neue Zertifikate erstellt werden müssen. Das Root-Zertifikat darf keinesfalls kompromittiert werden.

Die nachfolgende Grafik zeigt, wie die Zertifikate zum Signieren verschiedener Programmcodes und Konfigurationsdateien vom Root-Zertifikat abgeleitet werden und auch sogenannte „Certificate Revocation Lists“, CRLs, erstellt werden. Damit können im Bedarfsfall bei allen Geräten im Feld durch ein einfaches Update dieser Liste Zertifikate zurückgezogen bzw. ungültig gemacht werden.

Abbildung 6:
Darstellung eines
Zertifikatsbaums



- **Root-Zertifikate** sind unbegrenzt gültig. Sie werden sicher gegen Verlust aufbewahrt und werden nur benötigt, um neue Code-Signer-Root- oder Config-Signer-Root-Zertifikate zu erstellen. Bei Kompromittieren des Root-Zertifikats müssen die Geräte physikalisch getauscht werden.
- **Code Signer Root-Zertifikate** sind zeitlich begrenzt gültig. Sie werden verwendet, um die eigentlichen Bootsignier- und Codesignier-Zertifikate zu erstellen. Bei Verlust können neue Zertifikate mit dem Root-Zertifikat erstellt werden. Im Falle des Kompromittierens kann das Zertifikat auf eine Sperrliste (CRL) gesetzt werden oder wird durch Ablauf ungültig. Eine Sperrliste kommt beispielsweise in den Bootloader.
- **Boot Signer-Zertifikat** wird verwendet, um den Bootloader zu signieren.
- **Code Signer x-Zertifikat** wird verwendet, um bestimmte Betriebssystem-Images (Beispielsweise für VxWorks), oder Anwendungen zu signieren. Diese Zertifikate enthalten zusätzliche Parameter, in welchen Systemen sie gültig sind. Das Sperren erfolgt wie beim Code Signer Root-Zertifikat.
- **Config Signer Root-Zertifikat** wird verwendet, um Zertifikate zur Konfigurationsdatensignatur zu erstellen. **Config Signer x-Zertifikat wird verwendet**, um Konfigurationsdaten zu signieren.
- **CRL-Zertifikate** stehen für "Certificate Revocation List" und dienen zum Zurückziehen von Zertifikaten. Sie werden online oder über Updates verteilt.

Zusammenfassung

Die Integrität von Embedded-Systemen lässt sich durch die Anwendung kryptografischer Verfahren in einem klar definierten Ablauf und einer sicheren Hardware zur Schlüssel- und Zustandsspeicherung gewährleisten. Wibu-Systems bietet mit CodeMeter ein Smart Card basiertes Schutzsystem an, das für industrielle Schnittstellen verfügbar ist, gebräuchliche Betriebssysteme unterstützt wie Windows, Mac OS X, Linux sowie Windows Embedded, Real Time Linux, VxWorks, SPS-Systeme wie CODESYS und weitere. Es enthält eine sichere Implementierung symmetrischer und asymmetrischer Verschlüsselungsverfahren (AES, RSA, ECC) sowie Hash-Funktionen (SHA-256), Funktionen zur Signaturvalidierung (ECDSA) und einen Zufallszahlengenerator. Die im Weiteren verfügbaren Werkzeuge ermöglichen es, alle oben beschriebenen Schritte zum Integritätsschutz umzusetzen.

Abbildung 7: Schutzhardware CmDongles in verschiedenen Bauformen

Zentrale

WIBU-SYSTEMS AG
Rüppurrer Str. 52-54,
76137 Karlsruhe
Tel.: +49 721 93172-0
Fax :+49 721 93172-22
sales@wibu.com | www.wibu.com

WIBU-SYSTEMS Niederlassungen

WIBU-SYSTEMS (Shanghai) Co., Ltd.
Shanghai: +86 21 556 617 90
Peking: +86 10 829 615 60
info@wibu.com.cn

WIBU-SYSTEMS USA, Inc.
USA: +1 800 6 Go Wibu
+1 425 775 6900
sales@wibu.us

WIBU-SYSTEMS BV
Niederlande
+31 74 750 14 95
sales@wibu-systems.nl

WIBU-SYSTEMS NV/SA
Belgien | Luxemburg
+32 3 808 03 81
sales@wibu.be

WIBU-SYSTEMS LTD
Vereinigtes Königreich | Irland
+44 20 314 747 27
sales@wibu.co.uk

WIBU-SYSTEMS IBERIA
Spanien | Portugal
+ 34 91 123 07 62
sales@wibu.es

WIBU-SYSTEMS sarl
Frankreich
+33 1 86 26 61 29
sales@wibu.fr

WIBU-SYSTEMS AG, 1989 von Oliver Winzenried und Marcellus Buchheit gegründet und eigentümergeführt, ist ein innovativer Technologiepionier für Softwareschutz und Lizenz-Lifecycle-Management weltweit.

Als engagierter Anbieter einzigartiger, hochsicherer und flexibler Technologien hat Wibu-Systems mit CodeMeter eine umfangreiche, preisgekrönte Palette hard- und softwarebasierter Lösungen für PCs, Embedded-Systeme, Mobilgeräte, SPSen und Mikrocontroller entwickelt, die mit international patentierten Verfahren die Integrität digitaler Inhalte schützen.

Mit dem Motto „Perfection in Protection, Licensing and Security" hilft Wibu-Systems Softwareentwicklern und Herstellern intelligenter Geräte, ihr geistiges Eigentum in ihren Geräten und Anwendungen vor widerrechtlicher und unzulässiger Nutzung, Nachbau, Sabotage, Spionage oder Cyberangriffen zu schützen und gleichzeitig neue Geschäftsmodelle erfolgreich umzusetzen und in bestehenden ERP-, CRM- und E-Commerce-Plattformen zu integrieren.

SECURITY
LICENSING
PERFECTION IN PROTECTION

WIBU SYSTEMS

5062-002-01/20170222