Fraunhofer
AISEC

FRAUNHOFER RESEARCH INSTITUTION FOR
APPLIED AND INTEGRATED SECURITY

SCHUTZ EINGEBETTETER SYSTEME VOR PRODUKTPIRATERIE

TECHNOLOGISCHER HINTERGRUND UND VORBEUGEMASSNAHMEN

BARTOL FILIPOVIĆ, OLIVER SCHIMMEL

Fraunhofer
AISEC

Schutz eingebetteter Systeme vor Produktpiraterie

Technologischer Hintergrund und Vorbeugemaßnahmen

Bartol Filipović, Oliver Schimmel

15. November 2011

Version 1.1

Fraunhofer Research Institution for Applied and Integrated Security
www.aisec.fraunhofer.de

Fraunhofer AISEC
Parkring 4
85748 Garching (near Munich)
Germany

Kontakt: bartol.filipovic@aisec.fraunhofer.de

Inhaltsverzeichnis

Abbildungsverzeichnis

Bildnachweise

Bild auf Coverseite: fotolia/Martina Berg

Abbildung 1.1a (Seite 10): iStock-Foto/Aleksandar Ljesic
Abbildung 1.1b (Seite 10): iStock-Foto/Vladimir Ovchinnikov
Abbildung 1.1c (Seite 10): iStock-Foto/Alexander Shirokov
Abbildung 1.1d (Seite 10): iStock-Foto/Alexey Dudoladov
Abbildung 1.1e (Seite 10): Apple
Abbildung 1.1f (Seite 10): iStock-Foto/Jill Fromer

Abbildung 1.2 (Seite 11): Cfaerber, Wikimedia Commons, lizenziert unter Creative Commons-Lizenz Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Unported (CC BY-SA 3.0), URL: `http://creativecommons.org/licenses/by-sa/3.0/deed.de`. Die Originaldatei ist zu finden unter `http://upload.wikimedia.org/wikipedia/commons/d/d1/Geistiges_Eigentum_und_Wettbewerbsrecht.png`

Zusammenfassung

Die negativen Auswirkungen von Produktpiraterie sind eine ernstzunehmende Bedrohung. Davon betroffen sind neben den Originalherstellern auch übervorteilte Konsumenten und geschädigte Volkswirtschaften. Die Hersteller verlieren Marktanteile und erleiden Imageschäden. Konsumenten nutzen (unbewusst) minderwertige Produkte, deren Sicherheit und Zuverlässigkeit in Frage steht. Durch Investitionsrückgang, Arbeitsplatzabbau und Steuerausfälle werden letztlich sogar Volkswirtschaften geschwächt.

Eingebettete Systeme sind Bestandteil vieler moderner Investitions- und Konsumgüter. Wird auf vorbeugenden technologischen Schutz verzichtet, ermöglichen ausgefeilte Techniken Angriffe auf Hard- und Software in eingebetteten Systemen. Die Bandbreite der Angriffe reicht dabei vom gezielten Modifizieren bis hin zum vollständigen Reverse Engineering und Produktpiraterie. In diesem Bericht werden einerseits Angriffsmöglichkeiten und andererseits Schutzmaßnahmen erörtert, um der Produktpiraterie technologisch entgegenzuwirken.

Fraunhofer AISEC entwickelt auf Basis neuester wissenschaftlicher Erkenntnisse technologische Schutzmaßnahmen, um der Produktpiraterie entgegenzuwirken und Unternehmenswerte zu schützen. Die anwendungsorientierten Security-Spezialisten greifen dabei auf eine langjährige Projekterfahrung und fachlich ausgewiesene Expertise zurück. Das Angebot umfasst unter anderem:

- Modernes Labor für Hardwaresicherheitsanalysen und System-Evaluierungen
- Produktspezifische Sicherheitslösungen
- Hard- und Softwarebasierte Schutzmaßnahmen und Imitationsbarrieren (Schutz vor: Manipulationen, Reverse Engineering und Produktpiraterie)
- Komponenten- und Ersatzteilidentifikation
- Design Security
- Praxistaugliche Implementierung moderner Verschlüsselungstechniken

1 Produkt- und Know-how-Piraterie

Durch die unlautere Nachahmung von Produkten, Komponenten und Design entstehen große wirtschaftliche Schäden, die immer neue Rekordwerte erreichen. Die Geschädigten sind in erster Linie die Unternehmen, deren Produkte illegal nachgeahmt und vervielfältigt werden. Nachgeahmte Ware wird dann zu wettbewerbs- und marktverzerrenden Konditionen angeboten. Die negativen Auswirkungen reichen über den Verlust von Marktanteilen und Imageschäden bis hin zu Arbeitsplatzabbau. In unmittelbarer Folge wird damit auch die Volkswirtschaft direkt belastet. Aber auch Konsumenten sind von negativen Auswirkungen betroffen; falls beispielsweise einem Käufer anstatt eines vermeintlichen Markenprodukts eine minderwertige Fälschung untergejubelt wurde, deren Sicherheit und Zuverlässigkeit in Frage steht.

Schätzungen gehen davon aus, dass 10% des weltweiten Warenhandels von Produktpiraterie betroffen sind und sich der wirtschaftliche Schaden auf bis zu 300 Milliarden Euro beläuft [18]. Die Internationale Handelskammer (ICC) beziffert, in einer 2011 veröffentlichten Studie, den Gesamtumfang der Schattenwirtschaft mit Fälschungen und Raubkopien in den G20-Ländern für das Jahr 2008 sogar auf insgesamt 650 Milliarden US-Dollar [3]. Der auf Raubkopien von Musik, Filmen und Software entfallende Anteil schlägt darin mit 30 bis 75 Milliarden US-Dollar zu Buche. Durch negative gesamtwirtschaftliche Effekte entstehen Schäden von rund 125 Milliarden US-Dollar, darunter etwa Steuerausfälle, höhere Strafverfolgungskosten oder der Ausfall von Investitionen aus dem Ausland. Überdies würden laut der Studie mehr als 2,5 Millionen Jobs verloren gehen. Allein den jährlichen Verlust im deutschen Maschinen- und Anlagenbau schätzte der *Verband Deutscher Maschinen- und Anlagenbau* in einer 2010 veröffentlichten Studie auf 6,4 Milliarden Euro [16]. Tendenz steigend.

Die Bandbreite der durch Piraterie betroffenen Produkte ist groß. Betroffen sind sowohl Konsumgüter als auch Investitionsgüter. Beispiele für Imitate findet man in der Textil- und Kleidungsindustrie, bei digitalen Medien (Raubkopien von Musik, Filmen und Software), bei mechanischen Baugruppen und auch in der Elektronik (sowohl einzelne Bauteile als auch ganze Systeme). Durch moderne Verfahren zum Reverse Engineering und Rapid Prototyping stehen ausgefeilte Methoden und Werkzeuge für die Systemanalyse, Manipulationen und den Produktnachbau zur Verfügung [4, 13, 15].

Die Möglichkeiten zum Schutz von Produkten vor Piraterie sind ebenso vielfältig wie die betroffenen Produkte. Um der Piraterie entgegenzuwirken liegt es jedoch

nahe, sich mit Schutzmaßnahmen zu befassen, die auf die Sicherung des eigenen Kern-Know-hows zielen. Um sich wirksam gegen Plagiate zu schützen, müssen Unternehmen Vorsorge treffen, wobei ein wesentlicher Schwerpunkt beim Schutz von eingebetteten Systemen auf dem technologischen Bereich liegen sollte. Durch geeignete Maßnahmen kann dem Reverse Engineering oder der unerwünschten Manipulation an der Elektronik oder Software entgegengewirkt werden.

1.1 Begriffsbestimmung

Das Nachahmen von Produkten kann in verschiedene Kategorien eingeordnet werden, deshalb wird im Folgenden der Begriff Produktpiraterie definiert und auf die in diesem Artikel behandelten Systeme eingegangen.

1.1.1 Produktpiraterie

Umgangssprachlich werden neben dem Begriff Produktpiraterie auch andere Bezeichnungen synonym verwendet, wie etwa Produktfälschung, Plagiat, Produktimitation oder Markenpiraterie. Die Begriffsbestimmung nach [12] unterscheidet zunächst zwischen Imitation und Original: a) Die Imitation tritt zeitlich nach dem Original auf, b) die Imitation hat aus Sicht des Kunden eine ähnliche Anwendungsfunktionalität wie das Original, c) die Imitation basiert auf den gleichen oder sehr ähnlichen Technologien wie das Original, und d) die Imitation entsteht auf Basis illegitimer Nutzung fremden Technologie-Know-hows. Charakteristisch für Imitationen ist eine vollständige oder teilweise Nachahmung bestimmter Eigenschaften eines Originalprodukts. Spezielle Ausprägungen von Imitaten bilden nach [12] die beiden Klassen Plagiate und Fälschungen. Plagiate unterstellen fremdem geistigen Eigentum die eigene Urheberschaft. Fälschungen unterstellen einem eigenen Produkt unrechtmäßig die Urheberschaft eines Anderen. Der Begriff Produktpiraterie umfasst Fälschungen und Plagiate. Nach [18] handelt es sich bei Produktpiraterie um das verbotene Nachahmen und Vervielfältigen von Waren, für die der rechtmäßige Hersteller Erfindungsrechte, Designrechte und Verfahrensrechte besitzt.

1.1.2 Eingebettete Systeme

Bei einem eingebetteten System handelt es sich um ein elektronisches System (Computer), das in einen speziellen anwendungsorientierten Kontext eingebunden ist. Übliche Aufgaben sind Messen, Steuern, Regeln, das Überwachen von Daten oder die Signalverarbeitung. Das zugrundeliegende Computersystem ist sehr spezialisiert: Sowohl die Hardware als auch Software (Firmware) sind für ein bestimmtes Anwendungsszenario optimiert. Typische Randbedingungen dabei sind: minimale Kosten, geringer Platz-, Energie- und Speicherverbrauch und

(a) Digitalreceiver (b) Kartenterminal (c) Navigationssystem

(d) Spielkonsole (e) Smartphone (f) Tischwaage

Abbildung 1.1: Beispiele für eingebettete Systeme

die langfristige Einsetzbarkeit. Oftmals sind Echtzeitanforderung bei der Datenverarbeitung ein weiters wichtiges Kriterium. Beispiele für eingebettete Systeme sind in Abbildung 1.1 dargestellt.

1.2 Der Produktpiraterie entgegenwirken

Um seine Interessen zu schützen und der Produktpiraterie entgegenzuwirken, kann ein Originalhersteller sowohl organisatorische und rechtliche, als auch technologische Maßnahmen ergreifen.

Schutzmaßnahmen wirken entweder undifferenziert (für alle Produkte und Know-how-Bestandteile eines Unternehmens) oder differenziert (für ausgewählte Produktgruppen und Know-how-Bestandteile). Unternehmensstrategische Maßnahmen werden in [12] untersucht. Eine Darstellung der rechtlichen und organisatorischen Maßnahmen gegen Produktpiraterie enthält [17]. Erfahrungen mit der praktischen Anwendung von Schutzmaßnahmen sind in [9, 2, 1] beschrieben.

Bei den organisatorischen Maßnahmen handelt es sich beispielsweise um die Auswahl von Fertigungsstandorten, dem regulierten und gesicherten Know-how Zugang und um das Lieferketten- und Innovationsmanagement.

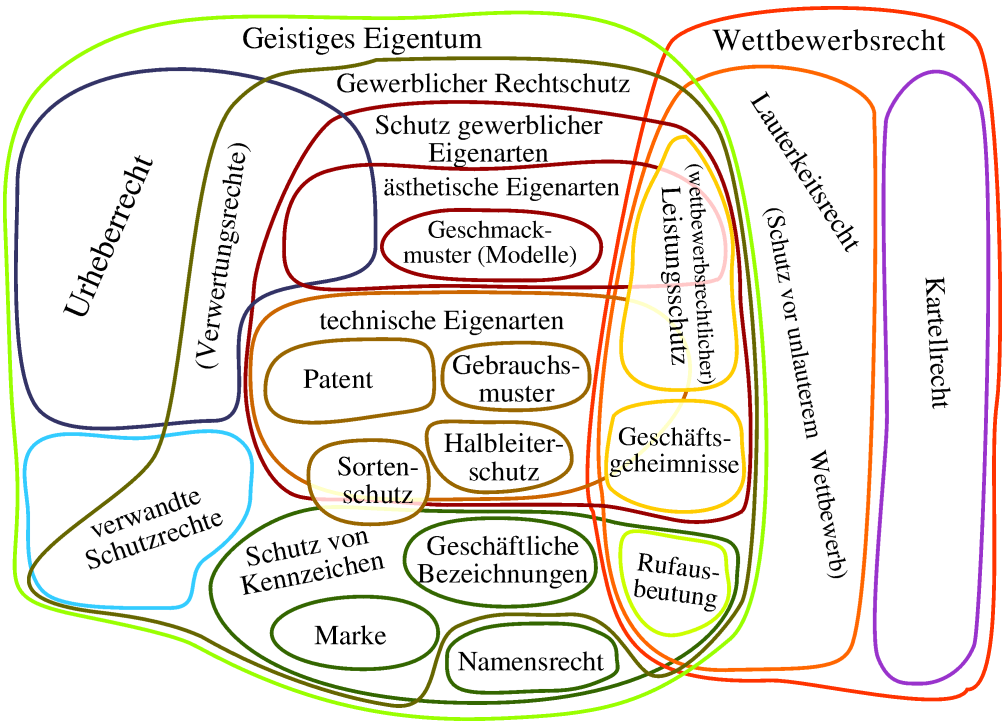


Abbildung 1.2: Verhältnis verschiedener Arten von geistigem Eigentum zum Wettbewerbsrecht

Zu den rechtlichen Maßnahmen zählt die Anwendung und Durchsetzung von gewerblichem Rechtsschutz (z. B. Patente, Markenrechte). Dabei ist zu beachten, dass die Unternehmensziele im Einklang mit dem Wettbewerbs- und Kartellrecht stehen müssen. D. h. auch die eventuell den eigenen Interessen entgegengesetzt stehenden Interessen von Verbrauchern und Wettbewerbern sind gesetzlich geschützt: Ein freier Marktwettbewerb ist erwünscht und einer Monopolbildung wird rechtlich vorgebeugt. Eine Übersicht relevanter Begriffe bezüglich des geistigen Eigentums und des Wettbewerbsrechts zeigt Abbildung 1.2.

Technologisch sind vielfältige Maßnahmen möglich, die auf unterschiedlichen Ebenen ansetzen können. Zum einen kann die unternehmenseigene Kommunikationsinfrastruktur gegen Industriespionage gesichert werden, um dem ungewollten Transfer von wertvollen Unternehmens-Know-how vorzubeugen. Dazu können Unternehmen geeignete Instrumente aus der IT-Sicherheit nutzen (z. B. Intrusion Prevention Lösungen oder Maßnahmen für Data Leakage Prevention). Zum anderen kann das Reverse Engineering von Produkten durch technologische Gegenmaßnahmen, die in die Produkte direkt integriert sind, erschwert werden. Es ist auch möglich durch geeignete Vorkehrungen den Schutz vor Manipulationen zu erhöhen. Zu den weiteren technologischen Möglichkeiten gehören auch Markierungs- und Kennzeichnungsverfahren zur Unterscheidung zwischen originalen und gefälschten Produkten. Einen Vergleich der wesentlichen Eigenschaften der beiden Bereiche *Markierung & Kennzeichnung* und *Anti Reverse Engineering* bietet Tabelle 1.1.

Um der Produktpiraterie effektiv entgegenzuwirken, müssen Angriffsszenarien

und Schutzziele aufeinander abgestimmt werden. Das Umsetzen geeigneter Schutzmaßnahmen muss dann schließlich praxistauglich, wirtschaftlich und rechtskonform sein.

	Markierung & Kennzeichnung	**Anti Reverse Engineering**
Zielgruppe der Maßnahme; (Aufwand liegt bei . . .)	Distributionsparteien (Zoll, Händler) & ggf. Kunde	Wettbewerber, Imitator
Prüfverfahren & Zielgruppen-Awareness erforderlich	Ja	Nein
Schutz vor produktspezifischen Know-how-Transfer und des geistigen Eigentums	Nein	Ja
Manipulationsschutz	Nein	Ja

Tabelle 1.1: Vergleich von technologischen Maßnahmen

1.2.1 Angriffsszenarien berücksichtigen

Die Angriffsszenarien sind produktspezifisch und können sehr unterschiedlich sein. Dementsprechend sind darauf abgestimmte Schutzmaßnahmen zu wählen. Beispielsweise besteht bei einer Spielkonsole die primäre Gefahr für den Hersteller nicht darin, dass das Produkt nachgebaut[1] wird, sondern darin, dass es durch Manipulationen an der Hard- oder Firmware sogar für Laien möglich wird raubkopierte Spiele zu nutzen [8, 14]. Ähnliches gilt für Navigationssysteme: Der Hersteller möchte aktuelles Kartenmaterial für seine Geräte verkaufen. Sein Interesse liegt also insbesondere darin, neben der Navigationssoftware auch sein Kartenmaterial vor Raubkopien zu schützen. Ein interessantes Beispiel sind manipulierte Kartenleser (Terminals) für Bankkarten: Es gab Fälle, bei denen Betrüger durch zusätzliche Hardware, die heimlich *in die Geräte integriert* wurden (Hardware-Trojaner), Geheimnummern und Kartendaten ausgespäht haben [6]. Die gewonnen Informationen wurden über das Mobilkommunikationsnetz als SMS-Textnachrichten an die Betrüger direkt weitergeleitet. Die dazu benötigte Mobilfunktechnik ist Bestandteil der trojanischen Hardware. Der Unterschied zu den sogenannten „Skimming"-Attacken besteht darin, dass die zusätzliche Hardware nicht einfach möglichst unauffällig von außen am Gehäuse des Terminals angebracht wird, sondern die spezielle trojanische Hardware direkt in die elektronischen Baugruppen des Kartenterminals integriert wird. Das kann sowohl innerhalb einer lückenhaft überwachten Produktion- oder Vertriebskette geschehen, oder auch durch gezielte Einbrüche in Verkaufsorte (Point of Sale), wo sich vielgenutzte Kartenterminals befinden.

Von Bedeutung für die Beurteilung der Pirateriegefährdung ist auch der Produktlebenszyklus eines Produkts. Zum Beispiel ist ein heute aktuelles Mobiltelefon nach einer gewissen Zeit (etwa zwei bis vier Jahre) für eine breite Kundenbasis

[1] Einem Produktnachbau wird in diesem Fall üblicherweise durch die Verwendung von proprietären und exklusiven Komponenten vorgebeugt (z. B. spezielle Grafikchips).

nicht mehr attraktiv, da die Technik schon als veraltet gilt. Für einen Produktpiraten bedeutet das ein verkürztes Zeitfenster, in dem er Imitate erstellen und anbieten kann. Andererseits gibt es Produkte, die einen deutlich längeren Lebenszyklus haben und in diesem Zeitraum fast unverändert angeboten werden (z. B. industrielle Anlagen oder Ersatzteile in der Automobilbranche).

Es ist plausibel anzunehmen, dass die Arbeitsweise von Produktpiraten bei ihrem Vorgehen produkt- bzw. komponentenspezifisch ist, d. h. die Methoden und Werkzeuge werden zielgemäß ausgewählt. Bei einem aus mehreren modularen Komponenten (etwa Gehäuse, Mechanik, Hardware, Elektronik, Software) zusammengesetzten Produkt kann der Produktpirat eine nützliche Kategorisierung der einzelnen Bestandteile durchführen: Welches sind Standardbauteile und wo befindet sich spezielles bzw. wertvolles Know-how, das zu extrahieren sich lohnt? Daraufhin kann entschieden werden, welche Nachbauvariante am besten geeignet ist: Das Übernehmen des Konzepts oder das Erstellen einer sklavischen eins zu eins Kopie.

Abhängig von der Komplexität des Systems können auch spezifische Strategien für einzelne Komponenten verfolgt werden. Die Betrachtung einzelner Kategorien oder Produktkomponenten ermöglicht einen reduktionistischen Lösungsansatz. Nach dem Grundsatz einer divide et impera Vorgehensweise lassen sich auch komplexe Teilkomponenten gezielt analysieren und verstehen, d. h. der Produktpirat kann den Aufwand auf interessante und werthaltige Komponenten fokussieren und ggf. Expertenwissen akquirieren. Dadurch kann auch die Gesamtheit eines Systems erfasst bzw. verstanden werden. Aus dem reduktionistischen Ansatz ergeben sich für den Produktpiraten mehrere Möglichkeiten für sein Vorgehen (Standardkomponenten werde als verfügbar angenommen):

Fokussierung und Spezialisierung: Eine Option ist die Verwendung gezielter und spezialisierter Methoden und Werkzeuge, mit denen ein Reverse Engineering relevanter Teilkomponenten möglich wird. Zum Beispiel könnten beim Nachbau von Gehäusen 3-D-Scanner verwendet werden, um Objektmaße für computergestützte Konstruktions- und Fertigungsverfahren zu ermitteln [13]. Auf Softwareebene wäre die Binärcodeanalyse ein Beispiel, für die je nach Rechnersystem spezielles Wissen, Vorgehen und Werkzeug erforderlich ist [4].

Substitution und Abwandlung: Widerstehen einzelne Komponenten dem Reverse Engineering, so kann eine alternative Technologie bzw. eine funktionell ähnliche Teilkomponente verwendet werden, die einfacher nachzubauen oder vielleicht schon als Standardtechnologie erhältlich ist. Es handelt sich also um eine Substitution von Technologien bzw. Komponenten. Dies kann dazu führen, dass der Nachbau technologisch bzw. qualitativ minderwertiger ist, da der Produktpirat eine von ihm nicht beherrschbare, hochwertigere Technologie durch eine andere ersetzt (beispielsweise Lehrbuchtechnologie). Die Qualitätseinbußen sind oftmals nicht ohne weiteres wahrnehmbar, d. h. das nachgebaute Produkt sieht dem Originalprodukt optisch ggf. sehr ähnlich und erscheint augenscheinlich funktionell gleichwertig. Anstatt eine exakte sklavische Kopien zu erstellen, kann es also das erklärte Ziel des Plagiators sein „lediglich ähnliche“ Plagiate

oder konzeptionelle Fälschungen herzustellen. Die gezielte Abweichung vom Originalprodukt kann sowohl die funktionalen, als auch die materiellen Eigenschaften des nachgebauten Produkts betreffen. Funktionale Abwandlungen schlagen sich mehr oder minder stark in den Produkteigenschaften nieder. Materialbetreffende Abweichungen hingegen können als Variante der Substitution betrachtet werden, die nicht unbedingt funktionale Auswirkungen haben müssen – trotz solcher Abwandlungen kann ein derartiger Nachbau u. U. die Kriterien für eine sklavische Kopie erfüllen, je nachdem wie streng die Bewertung erfolgt. Abwandlungen können selbstverständlich auch das äußere Erscheinungsbild oder das Bedienkonzept betreffen. Von besonderer Brisanz sind sicherheitsrelevante Abweichungen, die etwa die Betriebssicherheit negativ beeinflussen oder dazu führen, dass Sicherheitsnormen nicht mehr eingehalten werden können – und schlimmstenfalls gefälschte Gütesiegel den gegenteiligen Anschein erwecken.

1.2.2 Schutzziele bestimmen

Um das Vorgehen von Produktpiraten genauer zu verstehen, werden oftmals beschlagnahmte Imitate analysiert. Mit ihnen kann zum einen die Qualität der Produktnachbauten ermittelt werden und ggf. sind organisatorische bzw. technologische Vorgehensweisen zu erkennen. Zum anderen könnten auch Erkenntnisse zur Marktdurchdringung der Fälschungen und über die finanziellen Ressourcen der Produktpiraten von Belang sein. Aus den Vertriebswegen lassen sich ggf. Informationen über die organisatorischen Strukturen gewinnen.

Grundsätzlich sollte zur Vorbeugung von Produktpiraterie das Originalprodukt präventiv auf „Nachbau-Gefährdung“ überprüft werden. Es ist zu ermitteln, mit welchem Aufwand ein Nachbauen möglich ist. Ein Produktpirat wird abhängig von den Produkteigenschaften seine eigenen Zielsetzungen festlegen: Wie oben erläutert kann er versuchen interessantes Wissen aus dem Produkt zu extrahieren oder er ersetzt (falls möglich) komplizierte Bestandteile durch minderwertige Alternativen.

Am nützlichsten für das Kopieren eines Produkts erscheint eine methodische Analyse der Komponenten. Es findet eine Zerlegung des Produkts in seine Bestandteile statt, um funktionale oder technologische Abhängigkeiten zu erschließen. Dabei werden u. a. Standard- und Nichtstandard-Komponenten identifiziert.

Bei der Herstellung einer Fälschung stehen dem Produktpiraten unzählige Freiheitsgrade zur Verfügung, um vom Originalprodukt abweichende Veränderungen vorzunehmen. In diesem Zusammenhang ist es wichtig, die Möglichkeiten und Grenzen von potentiellen Schutzzielen und -maßnahmen richtig einschätzen zu können.

Unter Umständen können allgemeingültige Betrachtungen zu den Schutzmöglichkeiten der einzelnen Teilkomponenten gemacht werden. Zerlegt man das Produkt in geeigneter Weise in seine Grundteile (Produktpartitionierung), so erhält

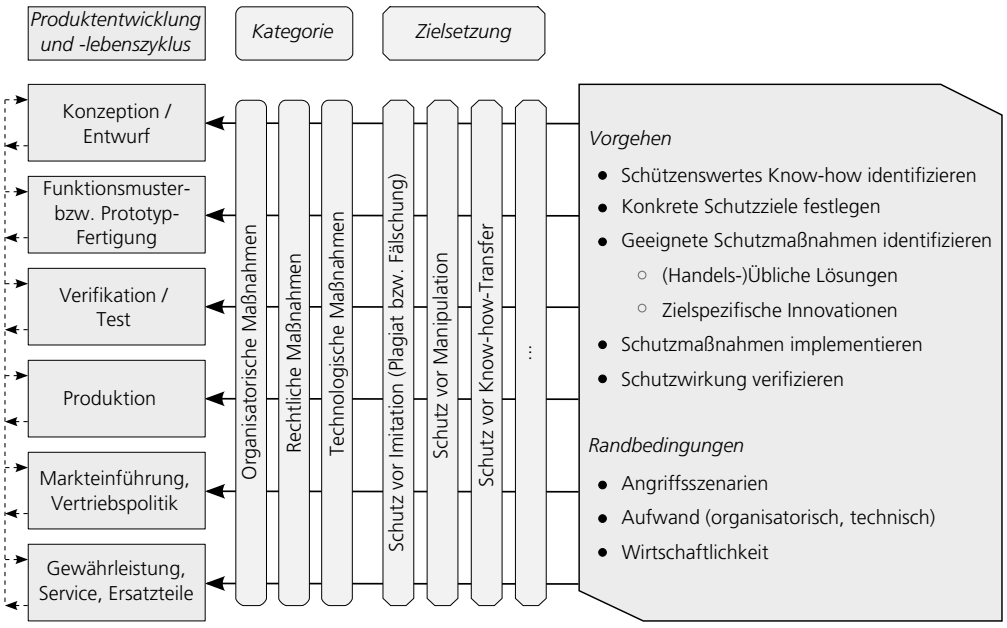


Abbildung 1.3: Schutzmaßnahmen im Produktlebenszyklus

man idealerweise eine nützliche Kategorisierung bezüglich des Schutzbedarfs einzelner Produktkomponenten. Maßnahmen sollten auf die im Produkt vorhandenen Unternehmenswerte und auf deren Schutz gerichtet sein. Eine entsprechende Fokussierung reduziert die Komplexität.

1.2.3 Schutzmaßnahmen umsetzen

Beim technologischen Schutz des Produkts können lokale, isolierte Maßnahmen ergriffen werden, aber auch ganzheitliche, die etwa Gehäuse oder Form des Produkts mit dem elektronischen Innenleben verknüpfen oder Hard- und Software kryptografisch miteinander verbinden. Es stellt sich die Frage, welche Maßnahmen die Gesamtheit eines Systems (das Gesamtprodukt) hinreichend gut gegen Plagiate schützt. Durch einen modularen Schutz ergeben sich unter Umständen lokale und isolierte Schutzmechanismen, die zielgerichtet umgangen oder ausgehebelt werden können. Durch Technologie- bzw. Komponenten-Substitution kann eine alternative Technologie bzw. eine funktional ähnliche Teilkomponente verwendet werden. Eine Konzeptkopie wäre damit realisierbar. Eine produktspezifische und ganzheitlich abgestimmte Wirkweise mehrerer Schutzmaßnahmen bietet ein relativ hohes Schutzniveau (siehe dazu auch Abbildung 1.3).

Die Technologieeigenschaften eines Produkts haben bedeutenden Einfluss auf das Gefahrenpotenzial. Besonders anfällig ist die weitverbreitete Standardtechnologie, die aus Gründen der Kostenreduktion verwendet wird. Typischerweise kann solche nichtexklusive Technologie von jedermann bezogen und verwendet werden. Einfache Lehrbuchtechnologie kann ohne weiteres nachgebaut oder

durch funktional ähnliche Technologie ersetzt werden. Lediglich eine umfassende (über die Standardtechnologie hinausgehende) Schutzmaßnahme scheint in diesem Fall von Nutzen zu sein. Im Idealfall bietet die Kerntechnologie per se bereits einen guten Nachbauschutz, weil sie schwierig zu beherrschen ist, spezielles Know-how benötigt und die Produkteigenschaften im wesentlichen von dieser Alleinstellungstechnologie abhängig sind. Ein Beispiel dazu ist in [9, Abschnitt 3.3.5] beschrieben.

Von Bedeutung ist sicherlich auch, wie aufwändig eine Know-how-Extraktion aus dem Produkt ist. Durch Reverse Engineering Methoden können ggf. sowohl Hard- als auch Software-Lösungen wertvolles Know-how preisgeben [13]. Auch hierbei kann die zugrundeliegende (zu analysierende) Technologie entscheidend für den Aufwand sein. Als konkretes Beispiel kann die Binärcodeanalyse dienen: Für die x86-Rechnerarchitektur gibt es diesbezüglich viele leistungsfähige (und zum Teil frei erhältliche) Werkzeuge (z. B. Disassembler, Debugger, Emulatoren, Analyse von Datei- und Speicherimages), wohingegen die Werkzeugverfügbarkeit für exotische Rechnerarchitekturen deutlich geringer ist – falls überhaupt vorhanden. Für konkrete Aufwands- und Kosten-/Nutzen-Einschätzungen ist technologische Expertise hilfreich. Auf dieser Grundlage kann beispielsweise beurteilt werden, ob bereits geeignete Werkzeuge zur Analyse verfügbar sind oder ob sie erst noch entwickelt werden müssen (was zeit- und kostenaufwändiger wäre).

Nützliche Schutzmaßnahmen beziehen sich unter Umständen auch auf die Produktionsverfahren, Produktionsorte und eventuell sogar auf die Vertriebswege. Durch geeignete organisatorische und ggf. technologische Maßnahmen kann man in diesen Bereichen konstruktiv und proaktiv eingreifen (Technologiedifferenzierung [9, Abschnitt 3.3.5], Track & Trace Technologie, Graumarktbekämpfung).

Die Schutzziele sind produktspezifisch zu formulieren und beziehen sich nur auf bestimmte Produkteigenschaften, Fertigungsverfahren oder Vertriebswege (vgl. Abbildung 1.3). Es ist festzulegen, welcher Schutz mit welchen Maßnahmen erreicht werden soll und es ist abzuschätzen, wie hoch die Nachbauhürde für einen Plagiator dadurch wird. Der Schutz wird sich auf die Kernkompetenz im Produkt erstrecken müssen, die das primäre Schutzziel sein sollte. Auch hierbei lassen sich Betrachtungen unterschiedlicher Feinheit anstellen. Wichtig ist, dass die Grenzen der gewählten bzw. umgesetzten Schutzmaßnahmen identifiziert werden (Evaluierung bzw. Verifikation des Schutzniveaus bezüglich der Schutzziele).

1.3 Fallbeispiel: Digitalreceiver Klon

Als weltweit ältester und größter Antennenhersteller und eines der führenden Unternehmen in der Kommunikationstechnik wurde die Firma Kathrein 2008 Opfer von Produktpiraten. Kathrein veröffentlichte dazu eine Dokumentation über den

Klon ihres Digitalreceivers UFS 910 [7], anhand derer sie auf die Unterschiede zwischen Original und Fälschung aufmerksam machten. Anhand dieses Fallbeispiels lässt sich die Vorgehensweise von Produktpiraten sehr gut veranschaulichen.

Die Täuschung des Verbrauchers beginnt in vielen Fällen mit der Imitation von Verpackung und Gehäuse. Dabei wird darauf geachtet, dass der Verbraucher die Fälschung möglichst ohne Bedenken als Originalprodukt identifiziert. Im Fall des Digitalreceivers wird genau auf diese Punkte Wert gelegt, es handelt sich also um eine sklavische Fälschung. Unterschiede zur Originalverpackung sind nur beim direkten Vergleich zu erkennen [7]: Aufkleber sind zum Teil direkt auf die Verpackung gedruckt, die aufgeklebte Seriennummer stimmt nicht mit der des Geräts überein und das Design entspricht der Vorgängerversion. Auch die Unterschiede am Gehäuse des Receivers und der Fernbedienung sind nicht ohne weiteres für den Verbraucher ersichtlich. Kleinigkeiten wie die Pigmentierung des Lacks oder unterschiedliche Anordnung der Lüftungslöcher fallen ohne den Direktvergleich nicht auf.

Waren vor einigen Jahren Fälschungen oftmals noch durch minderwertige Verarbeitung mit genauem Hinsehen erkennbar, steigt die optische Qualität mittlerweile soweit, dass die Kopie auf den ersten Blick nicht vom Original zu unterscheiden ist. Den Fälschern kommt dabei eine große Auswahl an Werkzeugen wie hochauflösende Kameras, 3-D-Scanner und Ähnliches zugute. Die Hersteller der Originalprodukte können das äußere Erscheinungsbild ihres Produktes kaum schützen. Eine Möglichkeit das Design seiner Verpackung oder des Gehäuses von Kopien unterscheidbar zu machen, ist das Kennzeichnen mit schwer zu fälschenden Codemustern oder elektronisch auslesbaren Etiketten. Hierzu zählen Hologramme und Kippfarben, deren Sicherheit auf der Geheimhaltung oder eingeschränkten Verfügbarkeit der Technologie beruht, Verschlusssiegel, die beim Öffnen des Produkts zerstört werden, die nur gering verfügbare Technologie der Mikroschrift (z.B. in Euro-Banknoten), 1-D-, 2-D- und 3-D-Barcodes, verschlüsselte Sicherheitscodes oder RFID-Transponder. Zudem gibt es Spezialfarben, Mikropartikel, DNA-Markierungsmoleküle, laserbasierte Oberflächenscans oder digitale Wasserzeichen, welche nur mit speziellem Messequipment geprüft werden können [2]. Diese Kennzeichnungen ermöglichen zwar das Erkennen gefälschter Produkte, verhindert allerdings nicht deren Herstellung und die Nutzung von fremdem geistigem Eigentum, wie schon in Tabelle 1.1 veranschaulicht.

Betrachtet man das Innere des Klons genauer, so fallen einige substituierte Standardbauteile auf [7]. Dies sind zum einen baugleiche, aber von anderen Herstellern bezogene USB- und CI-Slots, Netzschalter mit nicht DIN-EN-konformer Beschriftung und ein anderer Scart-Baustein. Zum anderen gibt es aber auch Bauteile, die zwar vom selben Hersteller kommen, aber nicht der originalen Ausführung entsprechen, wie das Display mit unterschiedlicher Zeichengröße.

Eventuell durch fehlendes Know-how entstandene Abwandlungen sind ebenfalls erkennbar. So ist eine Schutzfolie zwischen dem Netzteil und der RS232 Platine falsch montiert, welche im Original für das Erfüllen von Sicherheitsnormen verwendet wird. Im Originalnetzteil ist zusätzlich ein Deep-Standby Schaltkreis zur

Energieeinsparung integriert, dieser fehlt im Klon komplett. Funktional wirkt sich das Fehlen des Deep-Standby Schaltkreises von außen betrachtet nicht aus, qualitativ macht es den Klon allerdings minderwertiger.

In den folgenden Abschnitten soll nun weiter erläutert werden, wie Hard- und Software-Reverse Engineering den Fälschern die Möglichkeit eröffnet, den größten Teil der Hardware und sogar die auf dem Gerät installierte Software zu klonen und was ggf. dagegen unternommen werden kann.

2 Schutz von eingebetteten Systemen

Um ein angestrebtes Schutzniveau für eingebettete Systeme erreichen zu können, werden die Analyse- bzw. Schutzmaßnamen für Hard- und Software zunächst getrennt betrachtet. Die unterschiedlichen Ebenen der Maßnahmen überschneiden sich zwangsläufig an bestimmten Punkten, an denen Hard- und Software interagieren müssen. Für den Schutz des Gesamtsystems müssen daher Hard- und Softwaremaßnahmen aufeinander abgestimmt sein.

2.1 Hardware Reverse Engineering und Gegenmaßnahmen

Das Interesse ein fremdes System zu analysieren, genauer gesagt das Reverse Engineering, kann verschiedene Gründe haben. Abgesehen vom illegalen Nachbauen und Vervielfältigen von Originalware kann das Reverse Engineering für wirtschaftliche Aspekte verwendet werden, um beispielsweise abzuschätzen, welchen Wert ein Konkurrenzprodukt besitzt und welche Gewinnmarge ein Unternehmen damit am Markt erwirtschaftet. Zudem kann das gewonnene Wissen für die Weiterentwicklung eigener Produkte oder deren Vergleich mit der Konkurrenz herangezogen werden.

Möchte ein Unternehmen seine Produktpalette erweitern, so ist das Reverse Engineering eines schon auf dem Markt verfügbaren Produktes der Konkurrenz oftmals einfacher bzw. effizienter, als Zeit und Kosten in eigene Forschungs- und Entwicklungsarbeit zu investieren. Rechtlich gesehen befindet sich das Reverse Engineering von Elektronik in einer Grauzone zwischen legalem Analysieren von Konkurrenzprodukten, um sein eigenes Produkt kompatibel zu machen oder auch um Plagiate zu enttarnen, und dem illegalen Kopieren und damit dem Verletzen von Urheberrechten und Patenten.

Unternehmen wie die kanadische Firma Chipworks haben sich auf das Reverse Engineering spezialisiert. In einer Veröffentlichung aus 2009 [15] berichteten sie über aktuelle Vorgehensweisen beim Analysieren von Halbleiterelektronik. Dabei unterteilen sie das Reverse Engineering in folgende Schritte:

- *Produkt-Teardown* - Identifikation von Produkt, Package, internen Boards und Komponenten
- *System-Analyse* - Analyse von Funktionen, Timing und Signalpfaden
- *Prozess-Analyse* - Untersuchung der verwendeten Technologie
- *Schaltkreis-Extraktion* - Rekonstruktion der chipinternen Schaltung

2.1.1 Produkt-Teardown

Das Produkt-Teardown, oder anders ausgedrückt das Identifizieren einzelner Komponenten, stellt das oberste Level des Reverse Engineerings dar. Von Interesse sind dabei die verwendeten Bauteile einer Schaltung, um beispielsweise Materialkosten abzuschätzen. Das Board wird dazu abfotografiert und die Komponenten über ihren Gehäuseaufdruck identifiziert.

Verhindern lassen sich solche Analysen kaum, das Produkt-Teardown kann aber zumindest erschwert werden, indem man verwendete Beschriftungen abändert, sie nach der Boardbestückung weglasert (Laserradieren, siehe Abbildung 2.1) oder sie gar nicht erst aufdruckt. Zusätzlich kann die Schaltung auch mit undurchsichtigem Epoxidharz, Polyurethanharz oder Silikonkautschuk vergossen werden. Durch das Vergießen können allerdings Probleme mit Wärmemanagement und Wartbarkeit des Systems entstehen, da diese Techniken ursprünglich für Bauteile unter klimatisch schwierigen oder vibrationsintensiven Bedingungen konzipiert worden sind.

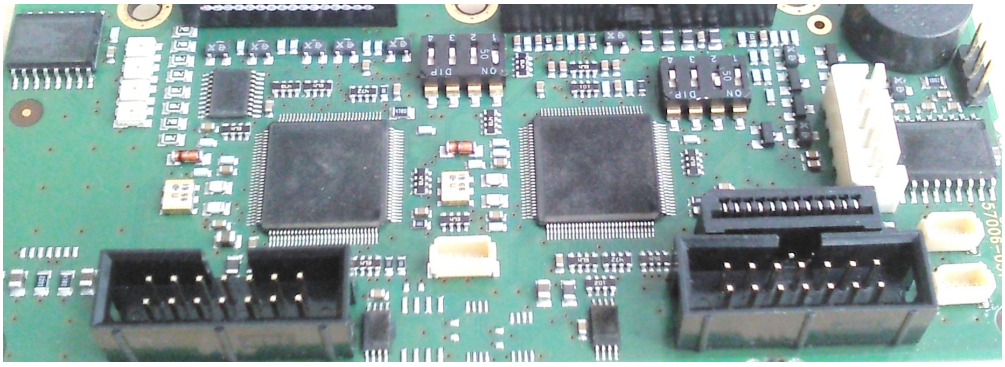


Abbildung 2.1: Entfernte Gehäusebeschriftung erschwert Chip-Identifikation

Diese Maßnahmen werden allerdings ausgehebelt, sobald dem Reverse Engineer ein Röntgengerät oder Werkzeug zum entfernen des Chipgehäuses (Ätzmittel, Schleifgeräte) zur Verfügung steht. Damit werden Chipbezeichnungen sichtbar, welche von den Halbleitermanufakturen zur Identifikation direkt auf dem Chip aufgebracht werden, wie eine Aufnahme eines aufgeätzten Mikrocontrollers in Abbildung 2.2 zeigt. Der vergrößerte Bildausschnitt links offenbart, um welchen Chip es sich handelt.

2.1.2 System-Analyse

Mit dem nächsten Schritt befindet man sich in der System-Analyse, bei der zusätzlich zur Komponentenidentifikation nun auch deren Kommunikation mit den anderen Bauteilen analysiert wird. Dabei kann man zwischen dem Reverse Engineering des Systems und der funktionalen Analyse unterscheiden. Beim Reverse Engineering des Systems wird wie beim Produkt-Teardown zunächst ein Foto von

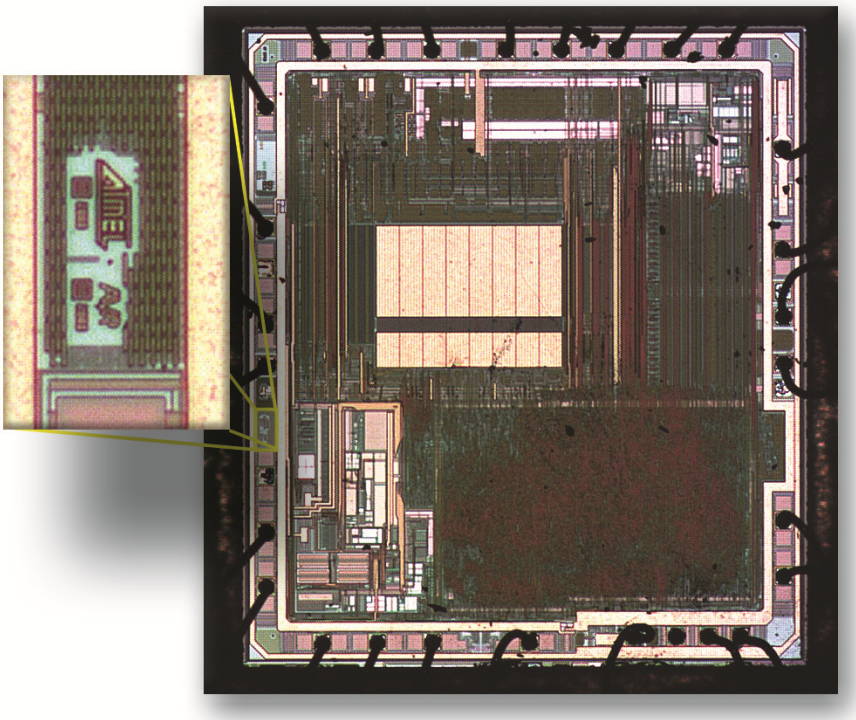


Abbildung 2.2: Entpackter Mikrocontroller offenbart Herstellerbezeichnung (Atmel AVR)

der Schaltung aufgenommen und die identifizierten Komponenten und deren Verbindungen notiert. Das Board wird somit Stück für Stück in seine Einzelteile zerlegt und analysiert. Bei Multi-Layer Boards wird zusätzlich jede einzelne Schicht abgetragen, digitalisiert und die Funktionalität des Systems über Softwaretools rekonstruiert.

Eine andere Vorgehensweise ist die funktionale Analyse, in der es nicht um die genaue Identifikation des Bauteils, sondern um dessen Funktionsweise geht. Dazu genügt in der Regel ein Signal-Generator, ein Logikanalysator und ein Oszilloskop [15]. Das System wird über den Signal-Generator mit bestimmten Mustern zu Operationen angeregt und die Funktionalität des Bausteins mit dem Logikanalysator und dem Oszilloskop ausgewertet. Diese Methode kann für die Substitution von Bauteilen verwendet werden.

Ein Erschweren der Analyse geschieht wie beim Produkt-Teardown durch entfernen von Chipbeschriftungen oder Vergießen der Schaltung, um die automatisierte Auswertung der Bilder mit Softwaretools zu verhindern. Das Nachvollziehen von Signalpfaden kann durch mehrschichtige Bauweise der Boards erschwert werden.

Nicht nur die Funktion eines Chips kann über die System-Analyse ermittelt werden, sondern auch die auf einem Mikrocontroller laufende Software oder die Implementierung eines FPGAs. Das Ausspähen lässt sich sowohl auf Software-, als

auch auf Hardwareebene erschweren. Softwareseitig kann der Code verschleiert oder die Lesbarkeit verkompliziert werden. Genaueres dazu wird im Abschnitt 2.2 erläutert.

Hardwareseitige Schwachstellen sind in erster Linie Speicherelemente zum Ablegen des Maschinencodes für Mikrocontroller oder die Implementierung eines FPGAs. Besonderes Augenmerk ist auf Speicherplatz für sensible Daten zu legen. In Security-Modulen sollten geheime Schlüssel nicht einfach aus dem Speicher oder über entsprechende Signalleitungen auslesbar sein.

Auslesen lassen sich diese Speicherelemente unter anderem über die eigentlich zum Programmieren und zur Fehlersuche bereitgestellten Programmier- und Debugging-Schnittstellen eines Bausteins, welche fast ausschließlich bei den Endgeräten vorhanden sind.[1] Immer häufiger sind daher in Bausteinen so genannte Soft- und Hardware-Sicherungen verbaut, die ein Auslesen des Speicherelements verhindern sollen. Dies wird zum Beispiel durch das Setzen bestimmter Bits im Baustein oder das Durchbrennen entsprechender Kontrollleitungen realisiert. Nicht immer sind diese Sicherungen resistent genug, um Manipulationen mit gezielten Verfahren Stand zu halten. Gesetzte Sicherungsbits können durch Laserbeschuss zurückgesetzt werden und durchgebrannte Leitungen durch IC-Mikrochirugie, Prozessanalyse und Schaltkreis-Extraktion lokalisiert und direkt auf dem Chip überbrückt werden. Ist die Sicherung zurück gesetzt, kann mit der System-Analyse fortgefahren werden. Effektive Gegenmaßnahmen sind Drahtgeflechte und reaktive Membranen, die um den Chip bzw. Baustein gelegt und elektrisch mit ihm verbunden werden. Wird nun das Chipgehäuse für die Schaltkreis-Extraktion geöffnet und damit die Drahtgeflechte bzw. Membranen zerstört, so hat dies ein Löschen der gesamten Bausteinfunktionalität zur Folge (siehe Abbildung 2.3).

Die rasant steigende Integration von programmierbaren Bausteinen in eingebetteten Systemen stellt gerade FPGA Hersteller vor die Herausforderung, Konzepte für einen effektiven Kopierschutz zu entwickeln. Grundsätzlich gibt es zwei Arten von FPGAs: zum einen die SRAM basierten, die beim Starten des Systems ihre Konfiguration aus einem externen, nicht-flüchtigen Speicher laden, zum anderen die nicht-flüchtigen FPGAs (Flash oder Antifuse), deren Konfiguration nach einmaligem Beschreiben bestehen bleibt. Bei SRAM basierten FPGAs muss darauf geachtet werden, dass die Konfigurationsdaten beim Laden in das FPGA nicht über den Datenbus ausgelesen werden können. Alle FPGA Hersteller setzen dazu auf proprietäre Konfigurations-Bitstreams, die ein Entziffern des zugrunde liegenden Codes erschweren bzw. verhindern sollen. Das Klonen des FPGAs wird dadurch allerdings nicht verhindert, da der Bitstream eins zu eins in ein Baugleiches FPGA eingespielt werden kann. Deshalb wird häufig der Bitstream zusätzlich verschlüsselt im Speicher abgelegt, beim Systemstart zum FPGA übertragen und erst dort wieder entschlüsselt. Dazu ist eine geeignete Schlüsselverwaltung notwendig, damit der Schlüssel nicht aus dem Chip ausgelesen werden kann. Eine neue

[1] Relevant sind beispielsweise: Joint Test Action Group (JTAG), Universal Asynchronous Receiver Transmitter (UART), Trace Analyser, In-System Programming (ISP).

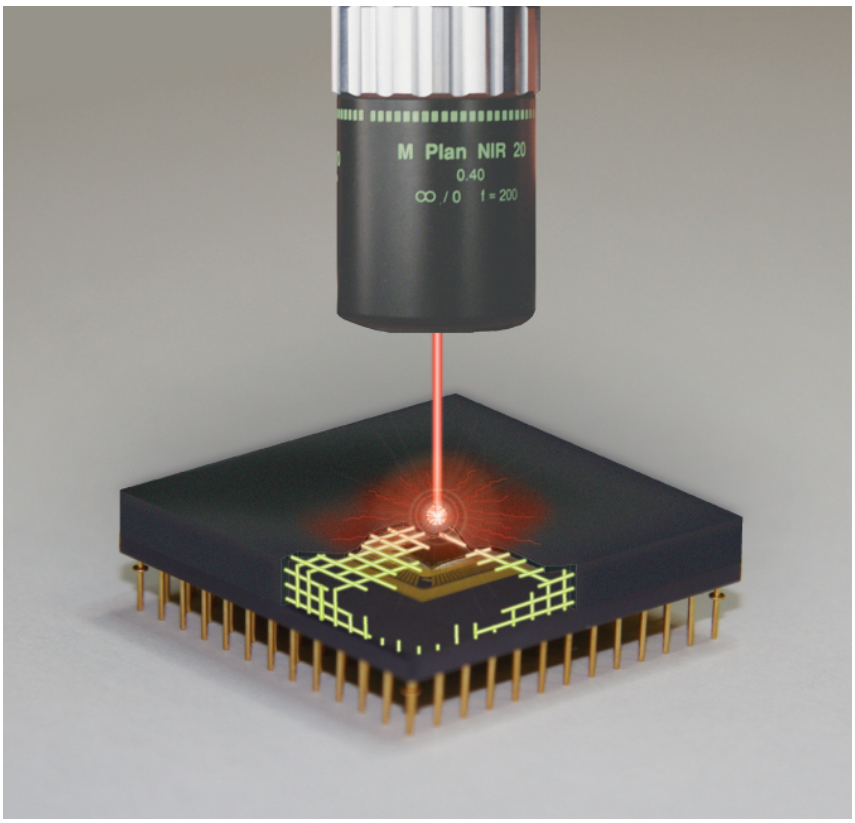


Abbildung 2.3:
Anti-Tamper Protection: Spezielles Chipgehäuse ermöglicht physische Angriffserkennung

Methode zur Schlüsselverwaltung wird derzeit am Fraunhofer AISEC in München untersucht. Dabei behilft man sich kleiner Ungenauigkeiten bei der Chipherstellung, die durch Prozessschwankungen unweigerlich entstehen, um daraus einen einzigartigen elektronischen Fingerabdruck zu generieren. Dieser Fingerabdruck kann dann als Schlüssel verwendet werden (siehe Abbildung 2.4).

Vorteil dieser Methode ist, dass der Schlüssel zur Laufzeit generiert werden kann und nicht in einem Speicher abgelegt werden muss. Zusätzlich lassen sich die Ungenauigkeiten der Schaltung nicht rekonstruieren und damit der Fingerabdruck nicht kopieren. Man spricht daher auch von einer Physical-Unclonable-Function (PUF). Seit 2002 beschäftigt sich die Wissenschaft mit der Thematik, wie PUFs in Schaltungen integriert werden können. Während dessen sind verschiedene Varianten zur Realisierung von PUFs entstanden, eine davon wird in [5] beschrieben, bei der eine PUF zusammen mit der dazugehörigen Messelektronik auf einem einzelnen Chip in Silizium integriert ist.

Das sichere Ablegen des Schlüssels alleine reicht allerdings in vielen Fällen noch nicht aus. Es kommt eine weitere Analyse hinzu, die es einem Angreifer ermöglicht den geheimen Schlüssel durch die Abstrahlung physikalischer Informationen während einer Verschlüsselung zu berechnen. Diese Analyse wird als Seitenkanalanalyse bezeichnet und bedient sich der physikalischen Größen Zeit und Leis-

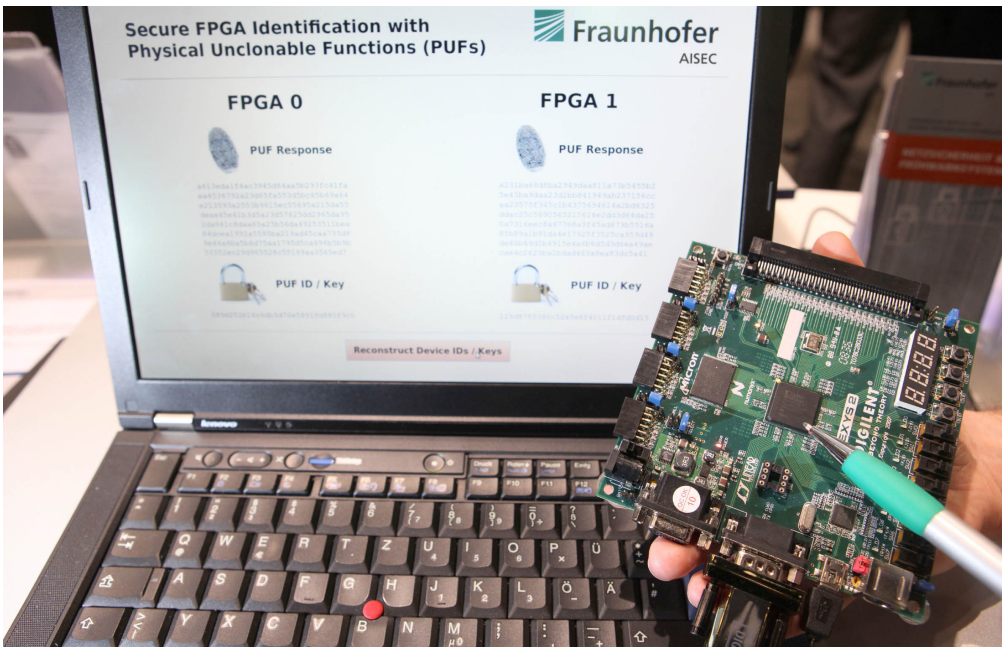


Abbildung 2.4:
Verwendung von *Physical Unclonable Functions* zur Generierung von geheimen Schlüsseln

tungsverbrauch. Eine „ungeschützte" kryptografische Schaltung kann beispielsweise in Abhängigkeit der zu verschlüsselnden Nachricht und des verwendeten Schlüssels unterschiedliche Berechnungszeiten benötigen, da datenabhängig verschiedene Funktionen ausgeführt werden oder unterschiedlich viele Speicherzugriffe erfolgen. Ist die Nachricht, wie zum Beispiel der verschlüsselte Bitstream für das FPGA bekannt, so kann über statistische Berechnungen der verwendete Schlüssel ermittelt werden. Im selben Stile lässt sich über den Leistungsverbrauch oder die elektromagnetische Abstrahlung des Bausteins der Schlüssel errechnen. Aktuelle Schaltungen werden fast ausschließlich in CMOS Technologie hergestellt. Diese Technologie verbraucht nur dann Strom, wenn die Transistoren im Chip von einem Zustand (0 bzw. 1) in den anderen Zustand (1 bzw. 0) ändern. Für die verschiedenen Kombinationen aus Nachricht und Schlüssel müssen jeweils unterschiedlich viele Transistoren ihren Zustand wechseln. Der Unterschied ist im Stromprofil erkennbar und gibt Informationen über den Schlüssel preis.

Neben den Seitenkanalanalysen kann man zudem die kryptografische Schaltung durch äußere Einflüsse (Wärmebestrahlung, ändern der Versorgungsspannung oder Taktfrequenz, Ladungsträgerinjektion durch Laserbeschuss, etc.) dazu zwingen, sich zu verrechnen (Abbildung 2.5). Der Schlüssel lässt sich dann durch Vergleichen der fehlerhaften und der richtigen Ausgänge ermitteln. Man spricht daher auch von einer Fehler-Attacke.

Es existieren bereits etliche Gegenmaßnahmen zum Erschweren von Seitenkanal- und Fehler-Attacken, die auf zufällig eingebauten Dummy-Operationen, Maskie-

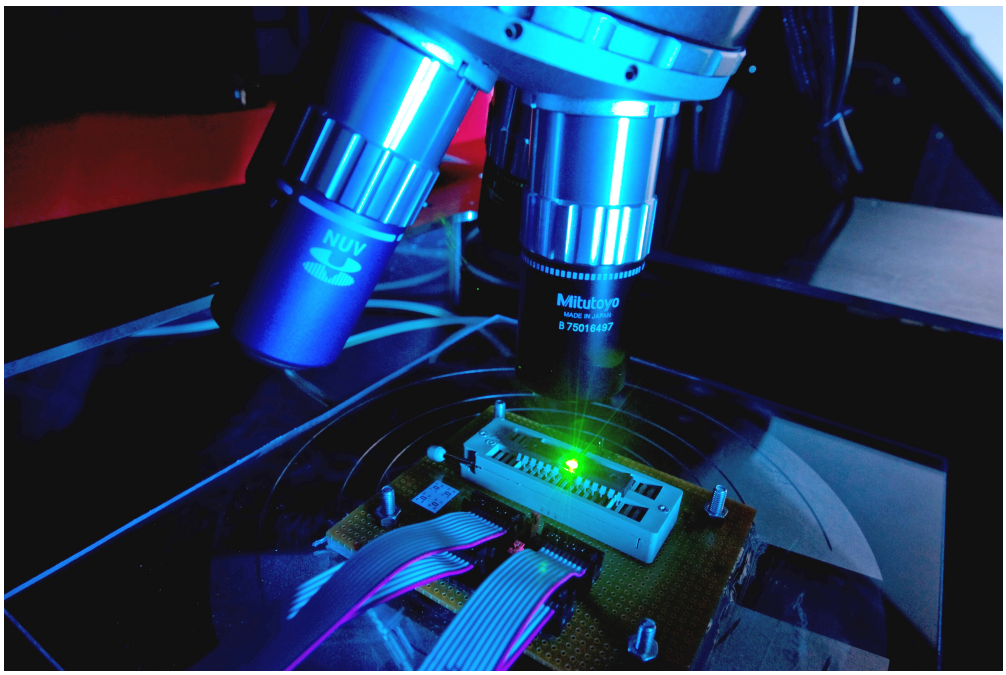


Abbildung 2.5: Fehler-Attacke durch Beschuss mit einem Laser

rung der Daten, Dual-Rail Technologie[2] oder auch eingebauten Rauschquellen basieren. Die physikalischen Sicherheitslücken und die damit verbundenen Gegenmaßnahmen sind allerdings immer Baustein- und Implementierungsabhängig. Daher befinden sich die Angriffsmethoden und die Gegenmaßnahmen in einem ständigen Wettstreit. Forschungsgruppen wie das Fraunhofer AISEC suchen daher stetig nach neuen Sicherheitslücken und entwickeln dafür spezifische Schutzmaßnahmen.

Es gibt spezielle Hardwarebausteine, die gegen die hier beschriebenen Angriffe gehärtet sind (vgl. auch [11]). In den aktuell weitverbreiteten und kostengünstigen Standardbausteinen sind allerdings meistens noch keine Schutzmaßnahmen integriert.

Für einige der hier vorgestellten Schritte in der System-Analyse sind Informationen über die verwendete Technologie der Bauteile notwendig. Hierfür bedient man sich der Prozessanalyse und der meist damit verbundenen Schaltkreis-Extraktion.

2.1.3 Prozess-Analyse

Informationen zu Prozessgrößen, Material und Aufbau eines Chips erhält man mit Hilfe der Prozess-Analyse. Analysemethoden und -werkzeuge sind weit verbreitet, da jede Halbleitermanufaktur diese zur Prozesskontrolle und Produktions-

[2] Einfügen einer zweiten Verschlüsselung, welche genau das entgegengesetzte Datum verarbeitet. Damit soll erreicht werden, dass der Leistungsverbrauch immer gleich ist.

fehleranalyse benötigt. Im ersten Schritt wird dazu das Gehäuse entfernt (Depackaging). Dies geschieht durch Ätzen in verschiedenen Säuren, bei Keramik-Gehäusen auch durch abschleifen. Ist der Chip erst einmal frei gelegt, können mit Hilfe verschiedener Mikroskope (Raster- und Transmissionselektronenmikroskop, Rastersondenmikroskop, etc.) und Chemikalien Materialübergänge, Strukturgrößen, Layeranzahl und p/n-dotierte Zonen sichtbar gemacht werden. Einen Schutz gegen diese Analyse gibt es nicht.

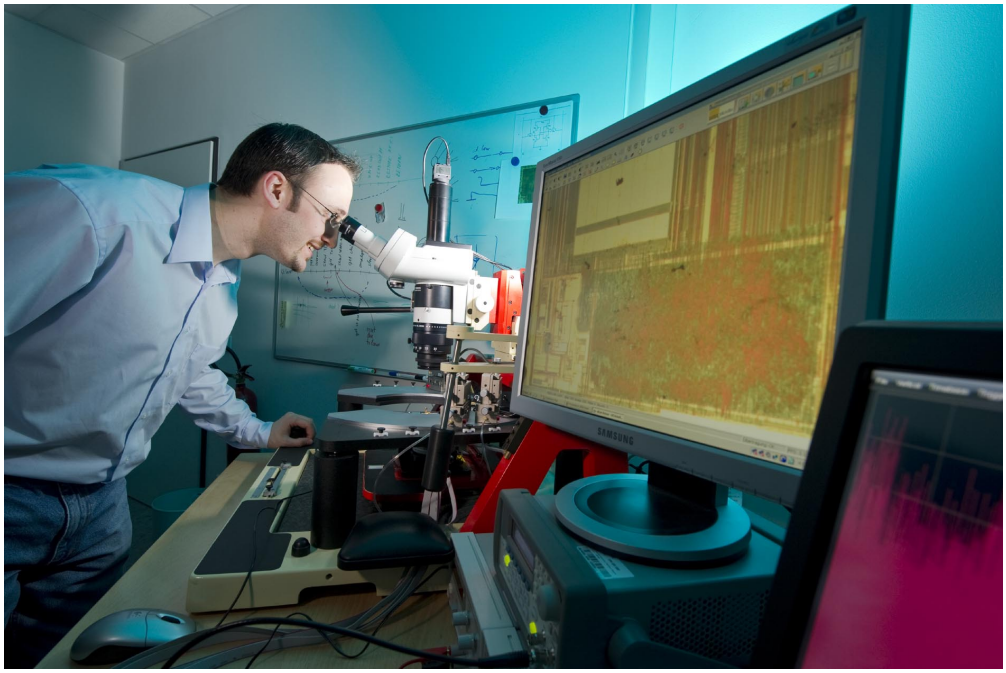


Abbildung 2.6: Mikroskop macht innere Schaltungen eines Mikrochips sichtbar

2.1.4 Schaltkreis-Extraktion

Nach dem Entfernen des Gehäuses und der aus der Prozess-Analyse gewonnenen Erkenntnis der einzelnen Schichtdicken kann Lage für Lage des Chips durch verschiedene Ätz- und Polierschritte abgetragen, abfotografiert und die erhaltenen Informationen schließlich über Software wieder zusammengefügt werden.

Nach den Aufnahmen folgt die eigentliche Analyse der Schaltung. Alle Transistoren, Spulen, Widerstände, Kapazitäten und Leitungen müssen in den einzelnen Schichten identifiziert und miteinander verbunden werden. Dies geschieht, je nach Komplexität der Schaltung entweder manuell oder mit der Unterstützung von Tools. Eine abschließende Verifikation überprüft, ob alle Komponenten verbunden und keine Vias (Vertical Interconnect Access - Durchkontaktierung) offen sind. So entsteht ein Schaltplan, welcher allerdings durch die heutige hohe Dichte an Funktionsblöcken sehr unübersichtlich ausfallen kann. Erfahrene Analysten untersuchen ihn daher genauer, identifizieren die Funktionsblöcke und vereinfachen damit die Darstellung.

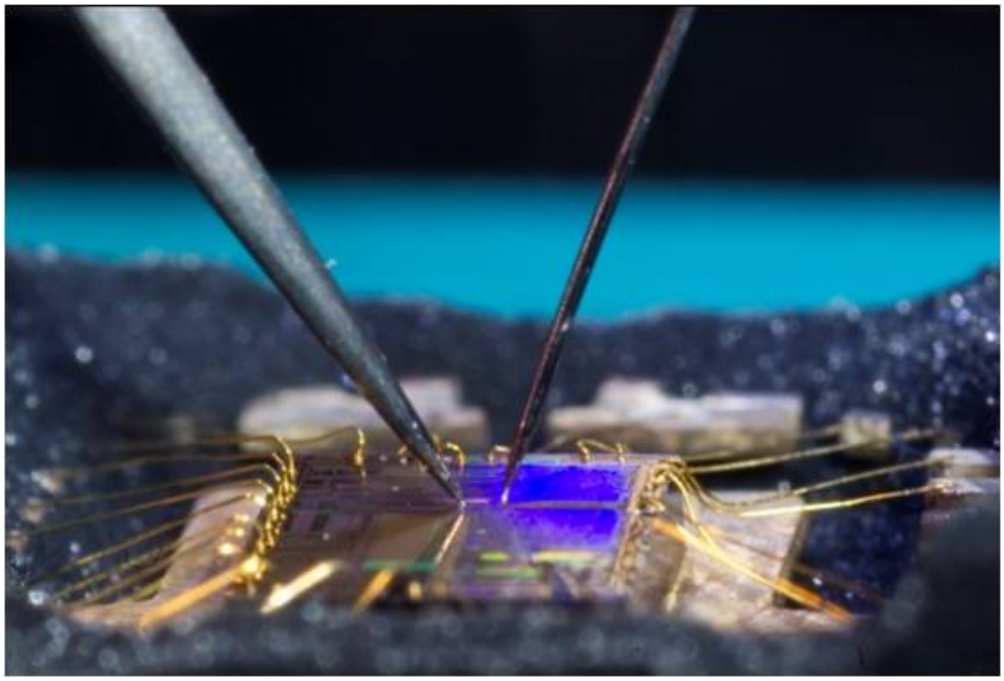


Abbildung 2.7: Mikroprobing in einem geöffneten Chip

Das Ergebnis dieser Analyse kann eine Liste von Schaltplänen für jede Schicht, eine vollständige Netzliste, Simulationen der Schaltung, Block- und Timing-Diagramme oder sogar Schaltungsgleichungen sein [15].

Ein Beispiel für die praktische Anwendung der Schaltkreis-Extraktion ist in [10] beschrieben.

2.2 Software Reverse Engineering und Gegenmaßnahmen

Kommt man in den Besitz des Maschinencodes eines Mikrocontrollers (Hexdump) oder der FPGA Implementierung, so besteht die Möglichkeit das gesamte in den Baustein investierte Know-how zu extrahieren. Bei Kenntnis des verwendeten Prozessors und dessen Befehlssatz lässt sich mit Hilfe von Disassemblern der Maschinencode in Assemblercode zurück verwandeln. Da Assemblercode meist nicht übersichtlich genug ist, gibt es zusätzliche Tools (Decompiler), die den Assemblercode in einen verständlicheren Pseudocode überführen (siehe Abbildungen 2.8, 2.9 und 2.10).

Gelingt es nichtflüchtige Speicher, wie etwa Flashbausteine, auszulesen oder anderweitig an Firmware-Code zu kommen (z. B. online zur Verfügung gestellte Update-Dateien), ist es durch Kenntnis des zugehörigen Dateisystems und Reverse Engineering oftmals möglich an wichtige Informationen, (Konfigurations-) Dateien oder geheime Parameter zu kommen. Diese Informationen ermöglichen

```
00402110  55 8B EC 83 EC 1C A1 84  6C 40 00 33 C5 89 45 F4  Uïýâý.íäl@.3+ëE¶
00402120  C7 45 FC 00 00 00 00 EB  09 8B 45 FC 83 C0 01 89  ÃE³....Ù.ïE³â+.ë
00402130  45 FC 83 7D FC 10 7D 11  8B 4D 08 03 4D FC 8B 55  E³â}³.}.ïM..M³ïU
00402140  FC 8A 01 88 44 15 E4 EB  E0 6A 00 8B 4D 0C 51 8D  ³è.êD.õÙÓj.ïM.Qì
00402150  55 E4 52 E8 B8 FD FF FF  83 C4 0C C7 45 F8 01 00  UõRÞ©²  â-.ÃE°..
00402160  00 00 EB 09 8B 45 F8 83  C0 01 89 45 F8 83 7D F8  ..Ù.ïE°â+.ëE°â}°
00402170  0B 73 36 83 7D F8 0A 73  0E 8D 4D E4 51 E8 BE F4  .s6â}°.s.ìMõQÞ¥¶
00402180  FF FF 83 C4 04 EB 0C 8D  55 E4 52 E8 10 F2 FF FF    â-.Ù.ìUõRÞ.=
00402190  83 C4 04 8B 45 F8 50 8B  4D 0C 51 8D 55 E4 52 E8  â-.ïE°PïM.QìUõRÞ
004021A0  6C FD FF FF 83 C4 0C EB  BB C7 45 FC 00 00 00 00  l²  â-.Ù+ÃE³....
004021B0  EB 09 8B 45 FC 83 C0 01  89 45 FC 83 7D FC 10 7D  Ù.ïE³â+.ëE³â}³.}
004021C0  19 8B 4D 10 03 4D FC 8B  55 FC 8A 44 15 E4 88 01  .ïM..M³ïU³èD.õê.
004021D0  8B 4D FC C6 44 0D E4 00  EB D8 8B 4D F4 33 CD E8  ïM³ãD.õ.ÙÏïM¶3-Þ
004021E0  E2 18 00 00 8B E5 5D C3  CC CC CC CC CC CC CC CC  Ô...ïÕ]+¦¦¦¦¦¦¦¦
```

Abbildung 2.8: Hexdump eines x86-Maschinencodes

auch gezielte Manipulationen an der Firmware vorzunehmen. So können ggf. Sicherheitsabfragen kompromittiert oder Ergänzungen an der Firmware vorgenommen werden. Beispiele aus der Praxis bietet das unberechtigte Entsperren von Computersoftware (Privilege Escalation, Rechteerweiterung) durch sogenannte Jailbreaks (etwa bei Smartphones oder Spielkonsolen).

Ist man erst einmal im Besitz des Codes, kann man ihn problemlos auf baugleiche Bausteine übertragen. Der Nachweis einer Code-Kopie ist nicht trivial, es existieren allerdings Ansätze, digitale Wasserzeichen (bekannt aus Bild- und Musikschutz) auf Codeebene zu realisieren oder über statistische Verfahren zu beweisen, dass die Wahrscheinlichkeit eines zufälligerweise erneuten Auftretens der gleichen Codefolge zu gering ist. Ist der Nachweis einer Kopie möglich, gibt es rechtliche Mittel, um gegen die Fälscher vorzugehen.

In vielen Ländern ist das Umgehen eines Kopierschutzes von Software und Implementierungen für kommerzielle Zwecke verboten. Neben dem Schutz vor illegalen Kopien des Softwarecodes möchten viele Unternehmen ihr in die Soft- oder Hardware integriertes Know-how nicht preisgeben. Die selbstentwickelten Verfahren oder Berechnungsvorschriften sind oftmals ein entscheidender Wettbewerbsvorteil und sollen daher geheim gehalten werden. Den Originalherstellern liegt demnach viel daran, die Schutzziele *Kopierschutz* und *Anti Reverse Engineering* für ihr Produkt durchzusetzen. Aus diesem Grund versuchen sowohl Softwareentwickler als auch Schaltungsdesigner vermehrt ihre Entwicklungen vor dem Ausspähen zu schützen.

Grundsätzlich gibt es keine ausschließlich auf Software basierende Methode, um Produkte vor dem Zugriff Dritter zu schützen. Ein wirksamer Schutz gegen das Ausspähen oder das Reverse Engineering von Softwareprodukten benötigt eine gewisse Hardwareunterstützung. Falls der Binärcode einer Software vollständig vorliegt, können auch darin enthaltene Schutzmaßnahmen analysiert werden, sofern diese lediglich softwarebasiert sind. Beispielsweise ist es wenig sinnvoll ein kryptografisches Verfahren und den zugehörigen Schlüssel im Softwarecode unterzubringen um damit andere Softwareteile abzusichern. Dies liegt u. a. daran, dass Software sich ohne Hardwareunterstützung nicht gegen die Veränderung,

```
.text:00402110 ; void __cdecl rijndaelEncrypt(const char *in, const char *expkey, char *out)
.text:00402110 _rijndaelEncrypt proc near              ; CODE XREF: _aes_encrypt_block128_key128+35↓p
.text:00402110                                         ; _do_test_crypt_aes+129↓p ...
.text:00402110
.text:00402110 state           = byte ptr -1Ch
.text:00402110 var_C           = dword ptr -0Ch
.text:00402110 round           = dword ptr -8
.text:00402110 i               = dword ptr -4
.text:00402110 in              = dword ptr  8
.text:00402110 expkey          = dword ptr  0Ch
.text:00402110 out             = dword ptr  10h
.text:00402110
.text:00402110                 push    ebp
.text:00402111                 mov     ebp, esp
.text:00402113                 sub     esp, 1Ch
.text:00402116                 mov     eax, ___security_cookie
.text:0040211B                 xor     eax, ebp
.text:0040211D                 mov     [ebp+var_C], eax
.text:00402120                 mov     [ebp+i], 0
.text:00402127                 jmp     short loc_402132
.text:00402129 loc_402129:                             ; CODE XREF: _rijndaelEncrypt+37↓j
.text:00402129                 mov     eax, [ebp+i]
.text:0040212C                 add     eax, 1
.text:0040212F                 mov     [ebp+i], eax
.text:00402132 loc_402132:                             ; CODE XREF: _rijndaelEncrypt+17↑j
.text:00402132                 cmp     [ebp+i], 10h
.text:00402136                 jge     short loc_402149
.text:00402138                 mov     ecx, [ebp+in]
.text:0040213B                 add     ecx, [ebp+i]
.text:0040213E                 mov     edx, [ebp+i]
.text:00402141                 mov     al, [ecx]
.text:00402143                 mov     [ebp+edx+state], al
.text:00402147                 jmp     short loc_402129
.text:00402149 loc_402149:                             ; CODE XREF: _rijndaelEncrypt+26↑j
.text:00402149                 push    0               ; rkidx
.text:0040214B                 mov     ecx, [ebp+expkey]
.text:0040214E                 push    ecx             ; expkey
.text:0040214F                 lea     edx, [ebp+state]
.text:00402152                 push    edx             ; state
.text:00402153                 call    _AddRoundKey
.text:00402158                 add     esp, 0Ch
.text:0040215B                 mov     [ebp+round], 1
.text:00402162                 jmp     short loc_40216D
.text:00402164 loc_402164:                             ; CODE XREF: _rijndaelEncrypt+97↓j
.text:00402164                 mov     eax, [ebp+round]
.text:00402167                 add     eax, 1
.text:0040216A                 mov     [ebp+round], eax
.text:0040216D loc_40216D:                             ; CODE XREF: _rijndaelEncrypt+52↑j
.text:0040216D                 cmp     [ebp+round], 0Bh
.text:00402171                 jnb     short loc_4021A9
.text:00402173                 cmp     [ebp+round], 0Ah
.text:00402177                 jnb     short loc_402187
.text:00402179                 lea     ecx, [ebp+state]
.text:0040217C                 push    ecx             ; state
.text:0040217D                 call    _MixSubColumns
.text:00402182                 add     esp, 4
.text:00402185                 jmp     short loc_402193
.text:00402187 loc_402187:                             ; CODE XREF: _rijndaelEncrypt+67↑j
.text:00402187                 lea     edx, [ebp+state]
.text:0040218A                 push    edx             ; state
.text:0040218B                 call    _ShiftRows
.text:00402190                 add     esp, 4
.text:00402193 loc_402193:                             ; CODE XREF: _rijndaelEncrypt+75↑j
.text:00402193                 mov     eax, [ebp+round]
.text:00402196                 push    eax             ; rkidx
.text:00402197                 mov     ecx, [ebp+expkey]
.text:0040219A                 push    ecx             ; expkey
.text:0040219B                 lea     edx, [ebp+state]
.text:0040219E                 push    edx             ; state
.text:0040219F                 call    _AddRoundKey
.text:004021A4                 add     esp, 0Ch
.text:004021A7                 jmp     short loc 402164
.text:004021A9 loc_4021A9:                             ; CODE XREF: _rijndaelEncrypt+61↑j
.text:004021A9                 mov     [ebp+i], 0
.text:004021B0                 jmp     short loc_4021BB
.text:004021B2 loc_4021B2:                             ; CODE XREF: _rijndaelEncrypt+C8↓j
.text:004021B2                 mov     eax, [ebp+i]
.text:004021B5                 add     eax, 1
.text:004021B8                 mov     [ebp+i], eax
.text:004021BB loc_4021BB:                             ; CODE XREF: _rijndaelEncrypt+A0↑j
.text:004021BB                 cmp     [ebp+i], 10h
.text:004021BF                 jge     short loc_4021DA
.text:004021C1                 mov     ecx, [ebp+out]
.text:004021C4                 add     ecx, [ebp+i]
.text:004021C7                 mov     edx, [ebp+i]
.text:004021CA                 mov     al, [ebp+edx+state]
.text:004021CE                 mov     [ecx], al
.text:004021D0                 mov     ecx, [ebp+i]
.text:004021D3                 mov     [ebp+ecx+state], 0
.text:004021D8                 jmp     short loc_4021B2
.text:004021DA loc_4021DA:                             ; CODE XREF: _rijndaelEncrypt+AF↑j
.text:004021DA                 mov     ecx, [ebp+var_C]
.text:004021DD                 xor     ecx, ebp        ; cookie
.text:004021DF                 call    @__security_check_cookie@4 ; __security_check_cookie(x)
.text:004021E4                 mov     esp, ebp
.text:004021E6                 pop     ebp
.text:004021E7                 retn
.text:004021E7 _rijndaelEncrypt endp
```

Abbildung 2.9:
Durch Disassembling des Hexdumps entsteht das zu Abb. 2.8 gehörende Assemblerlisting

```
void __cdecl rijndaelEncrypt(const char *in, const char *expkey, char *out)
{
  char state[16]; // [sp+0h] [bp-1Ch]@3
  unsigned int v4; // [sp+10h] [bp-Ch]@1
  const unsigned int round; // [sp+14h] [bp-8h]@4
  int i; // [sp+18h] [bp-4h]@1
  int v7; // [sp+1Ch] [bp+0h]@1

  v4 = (unsigned int)&v7 ^ __security_cookie;
  for ( i = 0; i < 16; ++i )
    state[i] = in[i];
  AddRoundKey((unsigned int *)state, (const unsigned int *)expkey, 0);
  for ( round = 1; round < 0xB; ++round )
  {
    if ( round >= 0xA )
      ShiftRows(state);
    else
      MixSubColumns(state);
    AddRoundKey((unsigned int *)state, (const unsigned int *)expkey, round);
  }
  for ( i = 0; i < 16; ++i )
  {
    out[i] = state[i];
    state[i] = 0;
  }
}
```

Abbildung 2.10:
Zum besseren Verständnis kann das Assemblerlisting aus Abb. 2.9 durch decompilieren in ein Pseudocodelisting überführt werden

Emulation oder dem überwachten und kontrollierten Ausführen in einer virtuellen Maschine absichern lässt. Durch geeignete Hardwareunterstützung können bestimmte Softwareteile beispielsweise in einem dezidierten Hardwarebaustein kryptografisch abgesichert werden. Dabei befinden sich die benötigten kryptografischen Verfahren und Schlüssel stets innerhalb des Hardwarebausteins und sind im Binärcode der Software nicht vorhanden und können somit auch nicht aus der Software extrahiert werden – weder durch statische Analyse des Binärcodes noch durch Emulation der Software in einer virtuellen Umgebung. Bei dieser Konstellation bilden Hard- und Software eine funktionale Einheit und eine sinnvolle Veränderung oder Analyse der verschlüsselten Softwareteile ist ohne die spezielle Hardware nicht möglich.

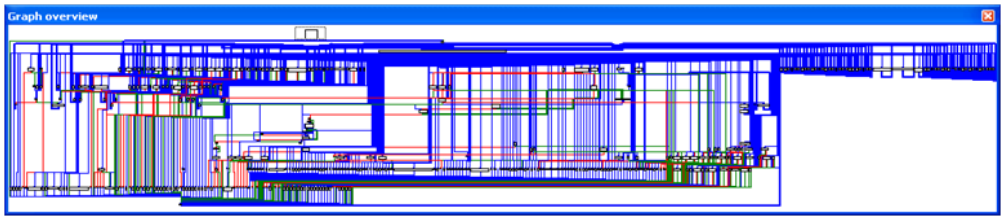
Auch hardwarebasierte Schutzmaßnahmen für Software sind nicht automatisch sicher. Als Beispiel dienen die ersten Dongle-Lösungen: die Zielsetzung ist, dass Softwareprodukte nur von autorisierten Benutzern eingesetzt werden können, also jenen die den Dongle physisch besitzen. Die Schwachstelle der ersten Dongle-Lösungen war das leichte Analysieren des Binärcodes der Software. Konnte man im Binärcode die Abfrage nach der Existenz des Dongels finden, so musste diese lediglich übersprungen werden, indem man beispielsweise den Assemblerbefehl *Branch Equal* in *Branch Not Equal* abänderte. Die derartig geänderte ausführbare Datei funktionierte dann ganz ohne Dongle. Moderne Lösungen sind nicht so einfach auszutricksen.

Man kann aus solchen Beispielen lernen, dass Softwareprodukte vor Manipulationen geschützt werden müssen, damit der Schutz durch die Hardware überhaupt

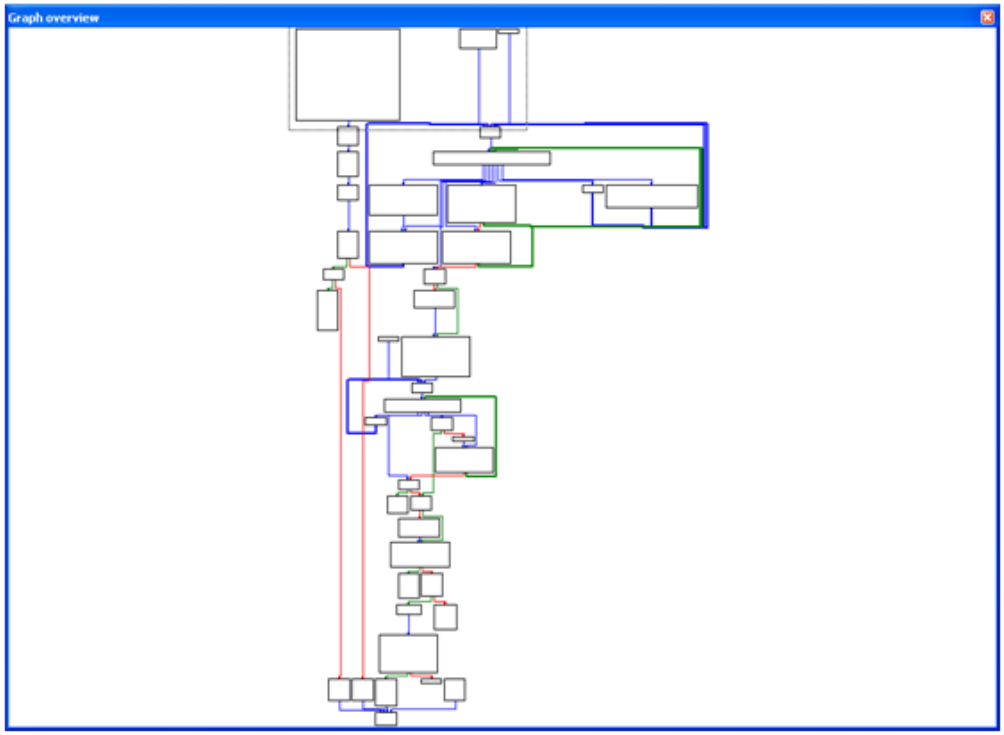
greifen kann. In der typischen PC-Umgebung kann dies nur schwerlich geleistet werden. Solche Systeme stellen aufgrund des Masseneinsatzes ein lohnendes Ziel für Cracker dar und sind gut dokumentiert und analysiert. Auch existieren vielfältige und ausgefeilte Tools für das Software Reverse Engineering und Manipulationen, die im Internet relativ leicht beschafft werden können. Ein Produkt, das auf Embedded Hardware beruht, kann leichter geschützt werden, weil dem Hersteller in der Regel mehr Optionen bei der Konfiguration der Hard- und Software zur Verfügung stehen, insbesondere falls alles aus einer Hand kommt und (sicherheitstechnisch) aufeinander abgestimmt werden kann.

Ein Softwareprodukt stellt in sich schon schützenswertes geistiges Eigentum dar. Daher kann schon ein Grundziel sein, dass der Angreifer nicht in der Lage ist, den Sinn und die Funktionsweise der Softwarekomponenten und internen Funktionen zu verstehen. Dazu werden sogenannte Obfuskatoren verwendet, welche den Code der Software verschleiern. Für Programmiersprachen wie Java und C# ist eine solche Schutzmöglichkeit wichtig, da diese Sprachen keinen nativen Maschinencode erzeugen, sondern die Quelldateien in eine Zwischensprache abbilden. Auf diese Zwischensprache lässt sich eine Binärcodeanalyse durchführen und somit aus den Assemblies des Microsoft .Net Frameworks durch entsprechende Klassenbrowser der ursprüngliche Quellcode rekonstruieren. Diese Eigenschaft ist im Hinblick auf die Produktpiraterie ein unerwünschter Zustand. Daher gibt es für Java und die .Net-Sprachen Programmpakete, die alle Funktionen, Variablen, Objekte, Klassen und Typen auf möglichst sinnfreie Namen umschreiben. Zusätzlich verschleiern diese Programmpakete die Aufrufe und Verknüpfungen zwischen verschiedenen Komponenten, verschlüsseln schließlich noch alle Zeichenketten und erstellen verworrenen Spaghetticode, damit der Angreifer möglichst wenig Anhaltspunkte für seine Analyse hat (vgl. Abbildung 2.11). Durch diese Maßnahmen sollen potentielle Angreifer abgeschreckt werden, sowie Kosten und Zeitaufwand für einen gewerbsmäßigen Angreifer schwer kalkulierbar und empfindlich erhöht werden. Allerdings ist eine eventuell nötige Fehlersuche nach der Transformation des Programms kaum noch realistisch durchführbar. Außerdem kann die Transformation durch den Obfuskator sogar zusätzliche Fehler im Softwareprodukt verursachen, die aufgrund der erschwerten Wartbarkeit möglicherweise erst beim Kunden auffallen und dann unvorhergesehene Kosten verursachen. Probleme ergeben sich unter Umständen auch durch das veränderte Laufzeitverhalten des Programms, einen erhöhten Speicherbedarf oder den erschwerten Umgang mit Updates oder Patches.

Auch (zum Teil proprietäre) Verschlüsselungs- oder Kompressionsverfahren finden bei der Absicherung von Software Anwendung. Zwar kann bei geeignetem Verfahren und fachmännischer Realisierung das Schutzniveau sehr hoch sein, allerdings geht dies ggf. einher mit Performanceeinbußen und einem recht hohem Know-how Bedarf für eine taugliche Umsetzung (diesbezügliche Stichworte sind: Schlüsselverwaltung, Auswahl von optimalen Verfahren, Implementierungsaspekte, Seitenkanalangriffe). Bei komplexen praktischen Anforderungen kann ein nichtspezialisierter Entwickler diesem Zusatzaufwand eventuell nicht gerecht werden.



(a) Verworrener Codefluss einer obfuskierten Funktion



(b) Einfacher Codefluss der obigen Funktion nach einer automatisierten Entwirrung

Abbildung 2.11: Auswirkungen einer (De-)Obfuskation auf den Codefluss

Die statische Analyse des Codes kann offenbaren, ob und wo kryptografische Berechnungen stattfinden. Entsprechende Stellen im Programm-Binärcode können durch charakteristische Codesignaturen gefunden werden (ähnlich zur Arbeitsweise von Virenscanprogrammen). Daher werden ergänzend zu den Standard-Verfahren teilweise proprietäre Verfahren genutzt oder es wird mittels Obfuskationstechniken versucht den Kryptocode zu verschleiern. Dynamische Codeanalysen und die Speicherüberwachung zur Programmlaufzeit können unter Umständen trotz Verschlüsselung zum Kompromittieren des Programms oder wichtiger Daten führen, wenn es gelingt an die relevanten kryptografischen Parameter im Computersystem zu kommen. Daher kombiniert man kryptografische Verfahren idealerweise mit spezieller Hardware.

Ohne spezielle Hardware ist es möglich den Binärcode anzugreifen (eventuell zur Ausführungszeit mit einem Debugger oder einer virtuellen Laufzeitumgebung). Dies zeigt sich auch an modernen Kopierschutzmechanismen für PC-Software, die in der Praxis erfahrungsgemäß analysiert und umgangen werden können.

Im Gegensatz zu den standardisierten PC Systemen bieten kundenspezifische eingebettete Systeme viel mehr Freiheitsgrade für die optimale Integration von Kryptoverfahren – und zwar in Soft- und Hardware, die sicherheitstechnisch aufeinander abgestimmt werden können.

2.3 Kombination von Hard- und Software Schutz

Ein Dongle (Kopierschutzstecker) dient primär dazu, Software vor unautorisierter Vervielfältigung zu schützen. Diese werden von bestimmten Herstellern zusammen mit einem Entwicklungspaket verkauft. Im Grunde genommen handelt es sich um ein *Hardware-Sicherheitsmodul* oder englisch *Hardware Security Module*, bei dem die kryptographischen Schlüssel sowohl gegen physikalische Angriffe (z.B. Seitenkanalangriffe) als auch softwarebasierte Attacken geschützt sind. Ein moderner Dongle für die USB-Schnittstelle hat etwa die Größe eines Memorysticks und enthält häufig die notwendigen Treiber in einem eigenen Speicher. Der interne Speicher ermöglicht die Integration von kryptografischen Verfahren direkt im Dongle. Durch Verwendung einer Verschlüsselung wird es einem Angreifer erschwert, die im Programm versteckten Abfragen an den Dongle zu lokalisieren und zu unterbinden. Über ein Verfahren zur Challenge-Response-Authentifizierung (Abbildung 2.12) kann die Dongle-Präsenz überprüft werden. Auch kann ein kryptografischer Dongle während der Programmlaufzeit eingesetzt werden, um die Software vor Manipulationen zu schützen (etwa durch eine kryptografische Integritätsprüfung der Binärcodeinstruktionen im Speicher) – zumindest wird die Reprogrammierung der Software deutlich erschwert. Daher ist die aktuelle Dongle-Generation bei gleichzeitig guter Implementierung nur noch sehr schwer zu überlisten. Durch die Verschlüsselung wird auch das geistige Eigentum vor fremden Augen gesichert, denn der Angreifer muss zunächst die Verschlüsselung überwinden, um seine Analysen beginnen zu können.

Ein *Secure Memory Device* funktioniert ganz ähnlich wie ein kryptografischer Dongle. Anstatt an einer externen Schnittstelle angeschlossen zu werden, wird ein Secure Memory Device schon im Hardwaredesign als Baustein mit integriert. Neben dem internen Speicher, der auch individuelle Parameter enthalten oder eine eindeutige Chip-Kennung haben kann, bieten diese Bausteine nützliche kryptografische Grundfunktionen, die zur Absicherung des Gesamtsystems (Hard- und Software) dienen (vgl. Abbildung 2.12).

In modernen PCs haben spezielle Bausteine unter der Bezeichnung *Trusted Platform Module* (TPM) ihren Einsatz gefunden, wo sie primär dazu verwendet werden, Manipulationen am BIOS und dem Ladeprogramm des Betriebssystems abzuwehren. Der Chip ist passiv und kann weder den Bootvorgang noch den Betrieb direkt beeinflussen. Er enthält eine eindeutige Kennung und kann daher zur Identifizierung des Systems verwendet werden. Als kryptografisches Device beherrscht der Chip kryptografische Verfahren und dient als sicherer Schlüsselspeicher und Zufallszahlengenerator.

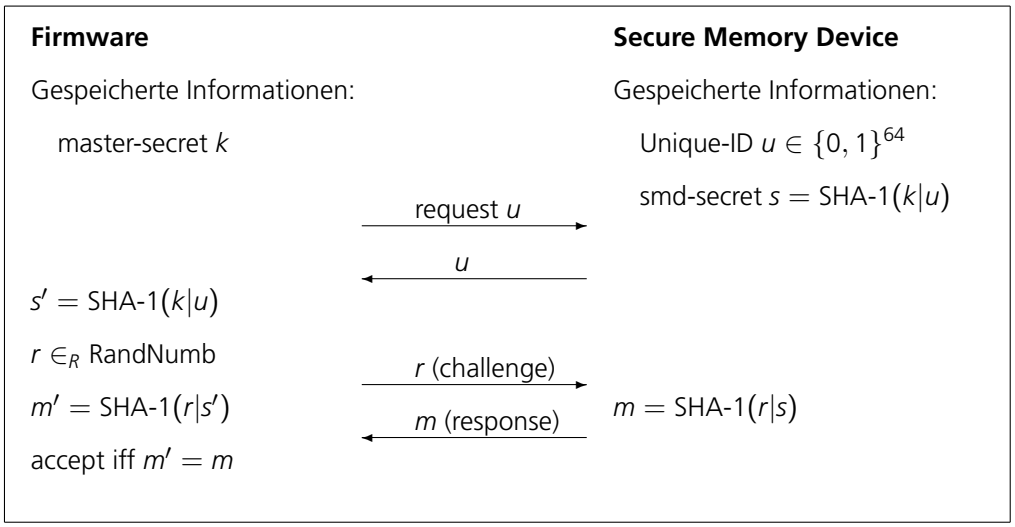


Abbildung 2.12:
Challenge-Response-Authentifizierung zur Überprüfung der Präsenz eines Hardwarebausteins

Der Großteil der Hersteller von programmierbaren logischen Schaltungen (etwa FPGAs) bieten Schutzmöglichkeiten für bestimmte Produktreihen an (*Design Security*). Manche verwenden hierfür ein zusätzliches Secure Memory Device. Einige FPGAs haben bestimmte Schutzmaßnahmen und / oder kryptografische Verfahren integriert.

Zu beachten ist, wie resistent die integrierten Bausteine gegen nicht-invasive und besonders auch gegen (semi-) invasive Angriffe sind. Bei lohnendem Erkenntnisgewinn sind manche Institutionen oder Konkurrenzunternehmen durchaus gewillt, die Kosten für die letztgenannten aufwändigen Angriffe zu zahlen und sich an kommerzielle Dienstleister mit gut ausgestatteten Laboren zu wenden, die auf das Reverse Engineering von Hardwarebausteinen und Platinen spezialisiert sind. Wie aufwendig die Rekonstruktion praktisch ist, hängt meist auch von der Fertigungstechnologie der Hardwarebausteine ab. Daher müssen die konkreten physikalischen Eigenschaften eines Bausteins mit in die Bewertung des Schutzniveaus einfließen.

Eine andere Möglichkeit geistiges Eigentum zu schützen ist die Fertigung von Spezialbausteinen, sogenannte ASICs. Diese Lösung ist durch die Rüstkosten sehr teuer und lohnt sich erst bei sehr großen Stückzahlen. Daher ist ein solcher Ansatz nur für Massenprodukte sinnvoll durchführbar. Aber auch hier gilt, dass der Aufwand für die Schaltungsrekonstruktion von den Fertigungsverfahren bzw. Strukturgrößen abhängt. Dies kann bedeuten, dass ein ASIC unter Umständen einfacher zu analysieren ist, als ein gegen mikroskopische Angriffe gesichertes FPGA. Die Sicherheit wird durch eine völlig gleichförmige mikroskopische Struktur solcher FPGAs erreicht, die auch nach der FPGA-Konfiguration erhalten bleibt. Diese Technologie bietet derzeit einen hohen Schutz gegen den Diebstahl von geistigem Eigentum.

Neben den bereits genannten Hardware-Bausteinen gibt es auch kryptografisch

befähigte Mikrocontroller und CPUs, die ggf. als Bausteine für die Umsetzung eines Schutzkonzepts dienen können. Es ist im Vorfeld der Verwendung solcher Bausteine zu klären, ob deren in Hardware realisierte kryptografische Verfahren dem Stand der Technik entsprechen und somit nicht durch Seitenkanalangriffe oder (semi-) invasive Verfahren kompromittiert werden können.

3 Profil Fraunhofer AISEC

3.1 Allgemeine Beschreibung

Als Spezialist für IT-Sicherheit entwickelt Fraunhofer AISEC unmittelbar einsetzbare Lösungen, die vollständig auf die Bedürfnisse der Auftraggeber ausgerichtet sind. Möglich werden diese maßgeschneiderten Dienste durch die mehr als 70 hochqualifizierte Mitarbeiter, die alle relevanten Bereiche der IT-Sicherheit abdecken. In München/Garching baut Fraunhofer AISEC derzeit drei neue Forschungs- und Entwicklungsbereiche auf, die sich mit hardwarenaher Sicherheit sowie dem Schutz von komplexen Diensten und Netzen befassen. Gleichzeitig entsteht ein Testzentrum für Sicherheits- und Zuverlässigkeitstests von Anwendungen und Komponenten (Hard- und Software) sowie Funktions-, Interoperations- und Konformitätstests.

3.1.1 Netzsicherheit & Frühwarnsysteme

Die Absicherung verteilter Netzdienste stellt für Unternehmen eine Herausforderung dar. Zum einen benötigen sie Mechanismen zur Identifizierung und Autorisierung der beteiligten Entitäten, zum anderen Mittel, um Integrität und Vertraulichkeit zu wahren. Zusätzlich müssen Unternehmen auch noch datenschutzrechtliche Bestimmungen einhalten. Der Bereich Netzsicherheit und Frühwarnsysteme entwickelt zusammen mit Industriepartnern Lösungen, die einen sicheren Betrieb von Diensten und Netzen ermöglichen. Hierzu erarbeitet Fraunhofer AISEC Konzepte, Verfahren und Protokolle, zum Beispiel in den Bereichen Sicherheit in All-IP & Future Internet, Personal Area Networks, Erkennung und Abwehr von Schadsoftware und Angriffen, kombinierte Techniken maschinellen Lernens, Bildverstehen und Sensorfusion.

3.1.2 Embedded Security & Trusted OS

Die Entwicklung von hardwarenahen Sicherheitslösungen muss bereits beim Design von Chips und Schaltungen ansetzen. Der Bereich Embedded Security & Trusted OS beschäftigt sich in diesem Kontext mit Verfahren zum Selbstschutz von Systemen durch gezielte System-Härtung und Verlagerung von Sicherheitsfunktionen in die Hardware. Um dies zu ermöglichen, entwickelt und erprobt Fraunhofer AISEC zusammen mit Industriepartnern vertrauenswürdige (mobile) Plattformen, fortgeschrittene Virtualisierungskonzepte, Verfahren zur Komponentenidentifikation sowie neue Testmethoden für eingebettete Komponenten. Am

Standort München wird dazu ein Hardware-Entwicklungs- und Prüflabor aufgebaut.

3.1.3 Sichere Services und Qualitätstests

Zusammen mit seinen Projektpartnern entwickelt der Bereich Sichere Services und Qualitätstests Lösungen und Testwerkzeuge für die Entwicklung, Komposition, Härtung, Bereitstellung und den Betrieb sicherer und verlässlicher Dienstebasierter Business-Software. Wichtig ist dabei besonders der Zuschnitt auf die besonderen Anforderungen der spezifischen Anwendungsdomänen wie Automobilbau, Logistik, E-Government (z. B. standardisierte Formate zum Datenaustausch in SOA-basierten Prozessen) oder das Gesundheitswesen. Der Bereich entwickelt zum Beispiel sichere Mehrwertdienste und Service-Plattformen für intelligente Umgebungen (Smart Factory, Smart Office), Sicherheitskomponenten, Risiko- und Compliance-Management-Verfahren sowie Web- Services unter Nutzung von Enterprise-Service-Bus-Infrastrukturen.

3.2 Leistungsangebot Produkt- und Know-how-Schutz

Fraunhofer AISEC kennt den Stand-der-Technik im Marktsegment der technologischen Schutzmaßnahmen. Im Kundenauftrag ermittelt das Institut, wie sich das jeweilige Produkt am besten gegen Imitationen schützen lässt.

Ungeschützte Komponenten in Produkten machen den Nachbau unnötig leicht und bedrohen die Investitionen innovativer Unternehmen. Fraunhofer AISEC zeigt Risiken auf und unterstützt Hersteller von eingebetteten Systemen bei der Gestaltung von Piraterie-robusten Produkten. Unternehmen werden bei der Auswahl geeigneter Schutzmaßnahmen und der optimalen Integration in ihre Produkte unterstützt.

Dort, wo die Schutzanforderungen durch marktübliche Maßnahmen nicht erfüllt werden können, etwa aus Performanz- oder Fertigungsgründen, entwickelt das Institut innovative Verfahren – etwa zum Schutz von eingebetteten Systemen. Dabei handelt es sich um wirkungsvolle und individuell anpassbare Lösungen, die einen hohen Schutz gegen Reverse Engineering bieten und proaktiv die Funktionsweise eines damit geschützten Produkts verschleiern. Dadurch wird ein Produktnachbau erschwert und wertvolles Unternehmens Know-how gesichert.

Bei der Realisierung werden produktspezifische Randbedingungen einbezogen, zum Beispiel möglichst nahtlose Integration in bestehende Unternehmensprozesse und Fertigungsverfahren. Bereits verfügbare Lösungen des Instituts sind als portable C Implementierungen realisiert und für Low-Cost Mikrocontroller Architekturen geeignet. Zudem werden für programmierbare Logikbausteine hardwarebasierte Referenzdesigns angeboten (auf Low-Cost FPGAs synthetisierbarer

VHDL-Code). Die Soft- und Hardware Lösungen sind miteinander kombinierbar und durch Parameter kundenspezifisch individualisierbar.

Darüber hinaus testet Fraunhofer AISEC eingebettete Systeme und IT-Prozesse auf Schwachstellen und Piraterie-Robustheit – Entwürfe und Prototypen ebenso wie fertige Produkte. Komplette Systemchecks sind genauso möglich wie Analysen ausgewählter Komponenten. Die Erfahrungen des Instituts im Bereich IT-Sicherheit und Design Security ermöglichen effektive Tests, deren Ergebnisse unmittelbar in Weiterentwicklung und Produktpflege einfließen können.

Themenbereiche:

- Modernes Labor für Hardwaresicherheitsanalysen und System-Evaluierungen [1]
- Produktspezifische Sicherheitslösungen
- Hard- und Softwarebasierte Schutzmaßnahmen und Imitationsbarrieren (Schutz vor: Manipulationen, Reverse Engineering und Produktpiraterie)
- Komponenten- und Ersatzteilidentifikation
- Design Security: Schutzmaßnahmen unter Verwendung von Hardwarebausteinen [2]
- Praxistaugliche Implementierung moderner Verschlüsselungstechniken
- Binärcodeanalyse (verschiedene Architekturen, etwa x86-, ARM und CIL-Code/.NET Bytecode)

Kompetenzen in den genannten Bereichen – für existierende oder erst in der Entwurfsphase befindliche Produkte:

- Analyse von Anwendungsszenarien, Bedrohungs- und Risikoanalyse
- Kundenspezifische Konzeption und Realisierung von innovativen Schutzmaßnahmen
- Unterstützung bei der Integration von etablierten Schutzmaßnahmen
- Proof-of-Concept und Prototyp Realisierung, Referenzimplementierung (Programmiersprachen bzw. Hardwarebeschreibungssprache VHDL oder Verilog)
- Analyse des Stands der Technik
- Erstellen von Marktübersichten zu kommerziellen Schutzmaßnahmen
- Technologieberatung und -bewertung, Maßnahmenempfehlung

[1] Getestet werden können sowohl einzelne Mikrochips als auch komplette eingebettete Systeme.

[2] Beispielsweise basierend auf FPGA, Secure Memory Device, Hardware Security Module, Trusted Platform Module oder kryptografisch befähigten Mikrocontrollern.

- Beurteilung von Schutzmaßnahmen (Schutzniveau), Bewertung des Nutzens (Abschätzen des Umgehungsaufwands)
- Gestaltung fälschungssicherer Produkte und Prozesse
- Test von Systemen und Prozessen auf Schwachstellen und Piraterie-Robustheit

Literaturverzeichnis

[1] Abele, E. (Hrsg.) ; Albers, A. (Hrsg.) ; Aurich, J.C. (Hrsg.) ; Günthner, A. (Hrsg.): *Innovationen gegen Produktpiraterie*. Bd. 3: *Wirksamer Schutz gegen Produktpiraterie im Unternehmen: Piraterierisiken erkennen und Schutzmaßnahmen umsetzen*. Frankfurt am Main : VDMA Verlag, November 2010. – 223 S. – Mit Ergebnissen aus den Projekten: ProOriginal, KoPira, KoPiKomp, ProAuthent. Weitere Bände der Reihe: [2, 9] 10, 40, 41

[2] Abramovici, M. (Hrsg.) ; Overmeyer, L. (Hrsg.) ; Wirnitzer, B. (Hrsg.): *Innovationen gegen Produktpiraterie*. Bd. 2: *Kennzeichnungstechnologien zum wirksamen Schutz gegen Produktpiraterie*. Frankfurt am Main : VDMA Verlag, November 2010. – 196 S. – Mit Ergebnissen aus den Projekten: MobilAuthent, O-Pur, EZ-Pharm. Weitere Bände der Reihe: [1, 9] 10, 17, 40, 41

[3] BASCAP: *Estimating the global economic and social impacts of counterfeiting and piracy*. London : Frontier Economics Ltd, February 2011. – URL `www.iccwbo.org/bascap` 8

[4] Eagle, Chris: *The IDA Pro Book: The unofficial guide to the world's most popular disassembler*. First Edition. No Starch Press, Inc., 2008 8, 13

[5] Gassend, Blaise ; Clarke, Dwaine ; Dijk, Marten van ; Devadas, Srinivas: Silicon physical random functions. In: *Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA : ACM, 2002 (CCS '02), S. 148–160 23

[6] Heise News: *Visa: Regelmäßiges Wiegen der Kartenterminals schützt vor Manipulationen*. Online News-Meldung vom 09.07.2010 12:16. 2010. – URL `http://heise.de/-1035169` 12

[7] Henke, Andreas ; Janssen, Mark: *Vergleich Kathrein UFS 910 Original mit China Klon*. Online Publikation (PDF-Datei). 2008. – Bebilderte Beschreibung der Unterscheidungsmerkmale zwischen original Kathrein UFS 910 und eines Imitats dieses Produktes 17

[8] Huang, Andrew: Keeping Secrets in Hardware: The Microsoft Xbox^{TM} Case Study. In: Kaliski, Burton S. Jr. (Hrsg.) ; Koç, Çetin Kaya (Hrsg.) ; Paar, Christof (Hrsg.): *Cryptographic Hardware and Embedded Systems – CHES 2002: 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*. Berlin, Heidelberg : Springer-Verlag, 2003 (Lecture Notes in Computer Science 2523), S. 213–227 12

[9] Kleine, O. (Hrsg.) ; Kreimeier, D. (Hrsg.) ; Lieberknecht, N. (Hrsg.): *Innovationen gegen Produktpiraterie*. Bd. 1: *Piraterierobuste Gestaltung von Produkten und Prozessen*. Frankfurt am Main : VDMA Verlag, November 2010. – 192 S. – Mit

Ergebnissen aus den Projekten: PiratPro, Protactive, Pro-Protect. Weitere Bände der Reihe: [1, 2] 10, 16, 40

[10] Krissler, Jan ; Nohl, Karsten ; Plötz, Henryk: Chiptease: Verschlüsselung eines führenden Bezahlkartensystems geknackt. In: *c't – Magazin für Computertechnik* 8 (2008), S. 80–85. – Heise Zeitschriften Verlag, Hannover 27

[11] National Institute of Standards and Technology: Federal Information Processing Standards Publication 140-2: Security Requirements for Cryptographic Modules / National Institute of Standards and Technology. 5 2001 (FIPS PUB 140-2). – U.S. government computer security standard. Last updated December 3, 2002 25

[12] Neemann, Christoph Wiard ; Klocke, F. (Hrsg.) ; Schmitt, R. (Hrsg.) ; Schuh, G. (Hrsg.) ; Brecher, C. (Hrsg.): *Berichte aus der Produktionstechnik*. Bd. 13: *Methodik zum Schutz gegen Produktimitationen*. 1. Auflage. Shaker Verlag, 2007 9, 10

[13] Raja, Vinesh (Hrsg.) ; Fernandes, Kiran J. (Hrsg.): *Reverse Engineering: An Industrial Perspective*. First Edition. London : Springer-Verlag, 2008 8, 13, 16

[14] Steil, Michael: *17 Mistakes Microsoft Made in the Xbox Security System*. Presented at the 22nd Chaos Communication Congress, December 29th 2005, 18:00, Berliner Congress Center, Berlin, Germany. 2005. – URL `http://events.ccc.de/congress/2005/fahrplan/events/559.en.html` 12

[15] Torrance, Randy ; James, Dick: The State-of-the-Art in IC Reverse Engineering. In: Clavier, Christophe (Hrsg.) ; Gaj, Kris (Hrsg.): *Cryptographic Hardware and Embedded Systems – CHES 2009: 11th International Workshop Lausanne, Switzerland, September 6-9, 2009, Proceedings*. Berlin, Heidelberg : Springer-Verlag, 2009 (Lecture Notes in Computer Science 5747), S. 363–381. – Invited Talk 3 8, 19, 21, 27

[16] VDMA: *VDMA-Umfrage zur Produkt- und Markenpiraterie 2010*. Verband Deutscher Maschinen- und Anlagenbau, 2010. – URL `www.vdma.org/produktpiraterie` 8

[17] Welser, Marcus von ; González, Alexander: *Marken- und Produktpiraterie: Strategien und Lösungsansätze zu ihrer Bekämpfung*. 1. Auflage. Wiley-VCH, 2007 10

[18] Wildemann, Horst ; Ann, Christoph ; Broy, Manfred ; Günthner, Willibald A. ; Lindemann, Udo: *Plagiatschutz: Handlungsspielräume der produzierenden Industrie gegen Produktpiraterie*. München : TCW Transfer-Centrum GmbH & Co. KG, 2007 8, 9