Oliver Schimmel, Maxim Hennig

Kopier- und Manipulationsschutz für eingebettete Systeme

Eingebettete Systeme sind heutzutage allgegenwärtig. Sie begegnen uns in Automobilen, Industrieanlagen, Versorgungsinfrastrukturen, der Unterhaltungselektronik und in der Medizin. Aber nicht jedes Gerät, das wie ein Original aussieht, ist auch ein Original. Um zu verhindern, dass – oft fehlerbehaftete – Nachbauten auf den Markt kommen und nicht nur dem Hersteller des Originals finanzielle Schäden und Imageverluste zufügen, müssen schon bei der Systemkonzeption Schutzmaßnahmen ergriffen werden. Der Beitrag stellt verschiedene Maßnahmen und deren Wirkungsweisen vor.

1 Einleitung

Eingebettete Systeme sind für besondere Anwendungen angepasste und optimierte Spezialsysteme. Meist müssen sie für die Erfüllung ihrer Aufgaben strengen Vorgaben folgen, wie beispielsweise einem geringen Stromverbrauch, niedrigen Materialkosten, einer kompakten Bauform oder einer garantierten Funktionalität unter Extrembedingungen. Man denke dabei an Steuergeräte im Auto, welche kompakt, zuverlässig und energiesparend sein sollen, gleichzeitig aber auch Wind und Wetter trotzen müssen.

Die Rahmenbedingungen für solche Produkte führen zu speziellen Designs, bei denen Einsparungen erfolgen, die fatalerweise meist zu Lasten von Schutzmechanismen ausfallen. Die Systeme werden dadurch anfällig für Produktpiraterie, bei welcher ein Fälscher das Originalprodukt entweder imitiert oder gar 1:1 nachbaut. Oftmals muss ein Fälscher dafür nicht einmal die exakte Funktionsweise des Gerätes im Detail verstanden haben [2]. Die notwendigen Aufwände für einen Nachbau unterschreiten in der Regel die Entwicklungs- und Vermarktungskosten des Originalproduktes um ein Vielfaches, wodurch die Fälschungen zu einem wesentlich günstigeren Preis angeboten werden können.

Je größer der Markt für ein bestimmtes Produkt wird, desto größer ist auch die Gefahr, dass das Produkt nachgebaut oder gefälscht wird. Gerade kleine und mittelständische Unternehmen trifft es oft besonders hart, wenn eines ihrer lang entwickelten Produkte gefälscht und zu einem günstigeren Preis auf den Markt gebracht wird. Viele Unternehmen realisieren diese Problematik leider erst, wenn sich die Fälschung bereits auf dem breiten Markt durchgesetzt hat und die finanziellen Schäden eingetreten sind. Andere Unternehmen hingegen erkennen immer häufiger schon im Vorfeld die Problematik und stellen den Schutz ihrer Systeme und insbesondere der darin enthaltenen Software immer mehr in den Vordergrund.

Der folgende Beitrag gibt einen Einblick in den Schutz eingebetteter Systeme vor Manipulation und Nachbau, im Speziellen dem Schutz vor den Methoden der Produktfälscher. Es wird zunächst das klassische Vorgehen eines Fälschers beim Analysieren des Originalproduktes beschrieben. Darauf abgestimmt werden danach entsprechende Lösungswege sowohl mithilfe von im Markt erhältlichen als auch mit aktuell noch in der Forschung befindlichen Komponenten wie dem PEP-Konzept (*Protecting Electronic Products*) des Fraunhofer AISEC vorgestellt.

Oliver Schimmel

wissenschaftlicher Mitarbeiter am Fraunhofer AISEC im Bereich Produktschutz. Forschungsschwerpunkt: Schutz integrierter Schaltkreise vor Manipulation und Nachbau.

E-Mail: oliver.schimmel@aisec.fraunhofer.de

Maxim Hennig

wissenschaftlicher Mitarbeiter am Fraunhofer AISEC im Bereich Produktschutz. Forschungsschwerpunkt: Schutz elektronischer Geräte vor Manipulation und Nachbau.

E-Mail: maxim.hennig@aisec.fraunhofer.de

2 Analysemethoden der Produktfälscher

Das Vorgehen eines Produktfälschers beginnt zunächst mit dem Identifizieren der Komponenten, die im System verbaut sind, dem so genannten Produkt-Teardown. Falls anhand der identifizierten Bauteile nicht bereits auf die gesamte Funktionalität des Systems geschlossen werden kann, wird ein Fälscher mit der System-Analyse fortfahren, bei welcher das Verhalten des Systems beobachtet

und die Kommunikation zwischen Bauteilen belauscht und analysiert wird. Besteht das System nicht allein aus Hardware-Komponenten, sondern verfügt zudem noch über eine entsprechende Steuersoftware, so wird ein weiterer Schritt des Fälschers sein, diese aus dem Originalsystem zu extrahieren, um sie – im Idealfall ohne jegliche Anpassungsnotwendigkeit – in seinen Nachbau einzuspielen [1, 2].

In der Software steckt oft jahrelange Entwicklungsarbeit und das eigentliche Kern-Know-how der Originalhersteller, weshalb sie ein beliebtes Angriffsziel für Produktfälscher darstellt und daher grundsätzlich Hauptbestandteil jeder Produktschutzmaßnahme sein sollte.

Das Auslesen der Software kann nun auf unterschiedliche Art und Weise geschehen. Beim Auslesen und Analysieren der Software im ausgeschalteten Zustand des Gerätes liest der Fälscher die Software direkt aus dem nicht-flüchtigen Speicher (zum Beispiel einem Flashbaustein) aus und analysiert sie mit Hilfe von kommerziellen oder frei erhältlichen Disassembler- und Decompiler-Tools [2]. Dieses Vorgehen bezeichnet man auch als statische Codeanalyse.

Ein Auslesen und Analysieren der Software im aktiven Betrieb des Systems erfolgt in der Regel durch Kontaktierung der freiliegenden Schnittstellen zum Debuggen und Programmieren, aber auch der Kommunikationsleitungen zwischen einzelnen Bausteinen. Analysiert man eine Software im laufenden Betrieb, so spricht man auch von einer dynamischen Codeanalyse.

Eine Anpassung der ausgelesenen Software vor dem Wiedereinspielen in die Fälschung ist notwendig, wenn beispielsweise Passwortabfragen in der Software integriert sind, die vom Fälscher entfernt oder übersprungen werden müssen.

Ein solches Auslesen und Analysieren von Software kann durch die Auswahl geeigneter Speicherbausteine mit integriertem Ausleseschutz und einer Verschlüsselung der Software erschwert werden. Der dafür notwendige kryptografische Schlüssel sollte dabei entsprechend sicher, versteckt und/oder verschleiert aufbewahrt werden. Einige ausgewählte Verfahren zum Schutz eingebetteter Systeme werden im folgenden Abschnitt erläutert. Weitere Verfahren sind in [2] zu finden.

3 Schutzmaßnahmen: Stand der Technik

Viele Produkte werden heute mit Schutzsiegeln oder Markierungen wie beispielsweise Hologrammen versehen, um ein Original von einer Fälschung abzuheben. Das Nachahmen oder Klonen elektronischer Produkte kann damit allerdings nicht verhindert werden, da ein Siegelbruch meist nur visuell und lokal nachgewiesen werden kann und eine fehlende Originalmarkierung erst entdeckt wird, wenn das Produkt bereits als Fälschung auf dem Markt erschienen ist. Eine Deaktivierung der Funktionalität des Produkts nach einem Siegelbruch ist zudem meist nicht realisiert. In der Regel finden solche Markierungslösungen daher eher in der Gewährleistung oder dem Markenschutz Anwendung. Der Fokus dieses Beitrags liegt nicht bei den rechtlichen Maßnahmen gegen Produktpiraterie, sondern konzentriert sich auf technische Lösungen, bei denen im Missbrauchsfall die Funktionalität des zu schützenden Gerätes deaktiviert wird.

3.1 Entfernen von Beschriftungen

Um einem Fälscher zunächst das Identifizieren der verwendeten Bauteile zu erschweren, können die Originalhersteller als einfache Gegenmaßnahme die Bauteilebeschriftungen mit einem Laser entfernen oder die Schaltung mit undurchsichtigem Epoxidharz, Polyurethanharz oder Silikonkautschuk vergießen [3]. Diese Maßnahmen lassen sich jedoch leicht umgehen, indem die Vergussmasse abgekratzt wird oder die Gehäuse der Bausteine mit chemischen Prozessen geöffnet und die auf dem Chip befindlichen Beschriftungen unter einem Mikroskop abgelesen werden [2].

3.2 Shielding

Eine andere Methode wird bei der Shielding-Technik verfolgt. Dort wird ein Schutzgitter aus Leiterbahnen über kritische Bereiche der Schaltung direkt auf dem Chip [4] oder über eine Leiterkarte hinweg gespannt [13,14]. Das Durchtrennen des Netzes wird von der darunter liegenden Schaltung erkannt, wodurch die Anwendung durch ein gezieltes Löschen der Software deaktiviert wird [5, 6, 7]. Dieser Schutz funktioniert allerdings nur, wenn Manipulationen im eingeschalteten Zustand des Gerätes erfolgen, da die Auswerteelektronik des Shieldings nur reagieren kann, wenn das System in Betrieb ist. So kann sich ein Angreifer den Zugang zu sensiblen Schnittstellen verschaffen, indem er das Gitter im ausgeschalteten Zustand öffnet, die interessanten Bereiche überbrückt und somit freilegt und erst danach das System wieder startet [8]. Das Auswertesystem geht dann davon aus, dass das Shielding noch intakt ist.

Komplexere Umsetzungen von Schutzgittern fragen hingegen nicht nur das Vorhandensein eines Gitters ab, sondern vergleichen die zur Laufzeit gemessenen physikalischen Eigenschaften des Schutzgitters, wie bspw. dessen Gesamtwiderstand, mit einem Referenzwert. Dieser Referenzwert muss dann allerdings an einer geeigneten Stelle im System abgelegt sein. Lässt sich dieser Wert oder dessen Abfrage auffinden und manipulieren bzw. überspringen, so kann das Gitter ohne Probleme entfernt werden. Daher ist es bei der Shielding-Technik gängige Praxis, dass für die zur Auswertung des Gitterzustandes notwendige Elektronik immer eine permanente Energieversorgung bereitgestellt wird, damit das System die Software bei einem Manipulationsversuch sofort löschen kann. Dies wird meist mit einer unter dem Shielding befindlichen Batterie bewerkstelligt, welche allerdings die Lebensdauer des Systems reduzieren kann.

3.3 Verwendung von Hardwareankern

Um eine Systemanalyse bzw. eine statische oder dynamische Codeanalyse zu erschweren, können Sicherheitsabfragen in die Software integriert werden. Dies können einfache Passwortabfragen oder auch Schlüsselabfragen aus so genannten Hardware-Ankern (ein Hardwarebaustein, welcher u. a. kryptografische Schlüssel und Funktionen sicher verwalten kann und welchem die Software „vertraut“) sein, wie ein *Trusted Platform Module* (TPM), ein Dongle oder ein *Secure Element*. Somit wird eine Bindung der Software an die Hardware realisiert und die Erzeugung einer direkten 1:1-Kopie des Systems erschwert. Diese Maßnahmen können zwar den Schutz erhöhen, ein Manipulieren des Systems aber zunächst nicht vollständig verhindern: So können Sicherheitsabfragen innerhalb der Software manipuliert und

übersprungen, Schlüsselabfragen aus Sicherheits-Chips abgehört und Dongle durch Emulatoren ersetzt werden [2].

3.4 Physical Unclonable Functions

Eine weitere – hauptsächlich noch in der Forschung befindliche – Schutzlösung für eingebettete Systeme bieten *Physical Unclonable Functions* (PUF) [9, 11, 12]. Sie basieren darauf, dass unkontrollierbare, nicht einmal vom Originalhersteller selbst reproduzierbare Fertigungstoleranzen eines Produktes ausgemessen werden, um daraus kryptografische Schlüssel oder Identifikationsnummern abzuleiten. Die Motivation für die Verwendung von PUFs ist die grundlegende Problematik, wie kryptografische Schlüssel oder Identifikationsnummern auslese- und manipulationssicher in einem System hinterlegt bzw. versteckt werden können. Mit Hilfe einer PUF kann auf das Abspeichern solcher Informationen verzichtet werden, da diese erst zur Laufzeit aus den Fertigungstoleranzen der PUF-Schaltung abgeleitet werden.

Ein Manipulationsschutz ist mit PUFs insofern gegeben, als dass sich die physikalischen Eigenschaften bei einem Eingriff (z. B. das Öffnen eines Gehäuses) in ein abgestimmtes System ändern und somit die extrahierten Informationen zur Generierung des Schlüssels verfälscht werden.

Aktuell beschränkt sich der Einsatz von elektronisch auswertbaren PUFs jedoch weitestgehend auf IC-Bausteine und umfasst keine gesamten Leiterplatten. Eine große Anzahl auf dem Markt erhältlicher Mikrocontroller oder Integrierter Schaltkreise (IC) sehen keinen Schutz durch die PUF-Technologie vor. Ein Nachrüsten ist zudem nicht praktikabel, da die Integration meist in das Design des gesamten ICs eingreifen würde und somit ein Re-Design des Chips notwendig wäre.

4 Schutz mit PEP

Die zuvor vorgestellten Lösungen zur Absicherung eingebetteter Systemen bieten einzelne und isolierte Maßnahmen wie die Manipulationsüberwachung (Shielding) oder die Bindung der Software an einen Hardwareanker (TPM, Dongle, etc.). Einen Ansatz für die Kombination aus Manipulationsüberwachung und Software-/Hardware-Bindung stellen PUFs dar. PUFs wirken allerdings nur passiv auf Manipulationsversuche ein, da ihnen grundsätzlich eine Laufzeit-Manipulationsüberwachung fehlt. So kann bspw. ein Angreifer einfach abwarten, bis die Software mit der intakten PUF entschlüsselt wurde, daraufhin – falls nötig – die PUF-Struktur entfernen und die entschlüsselte Software aus dem System auslesen.

Das im Folgenden beschriebene PEP-Konzept [10] soll nun verdeutlichen, wie eine geschickte Kombination verschiedener Schutzmaßnahmen ein System vor den hier beschriebenen Angriffsmethoden bewahren kann.

4.1 Schützenswerte Elemente identifizieren

Um die Sicherheit eingebetteter Systeme zu erhöhen ist es zunächst erforderlich, die kritischen und schützenswerten Bereiche des Systems zu identifizieren und daraufhin dem Angreifer den Zugriff darauf zu verwehren. Der Angreifer soll keinerlei Möglichkeit haben, aus den kritischen Zonen sensible Informationen extrahieren zu können. Zu den schützenswerten Bereichen gehören in erster Linie Bausteine des eingebetteten Systems, in welchen die Software gespeichert ist. Weiterhin gehören hierzu auch verschiedenste Arten von Schnittstellen, über welche sich die Software auslesen oder manipulieren lässt.

4.2 Umsetzung des PEP-Konzeptes

Das PEP-Konzept schützt kritische Bereiche des elektronischen Systems, indem diese mit einer elektronisch auswertbaren Schutzfolie abgedeckt und somit für einen Produktfälscher unzugänglich gemacht werden (siehe Abbildung 1 und Abbildung 2). Das Konzept erlaubt es, Manipulationen am System während des Betriebs mittels in der Schutzfolie integrierten Sensoren zur Impedanzmessung zu detektieren und darauf zu reagieren. Die dazu notwendige Mess- und Auswerteelektronik befindet sich dabei unterhalb der Folie. Versucht ein Produktfälscher die Folie zu öffnen oder zu entfernen, um sensible Daten aus dem System zu extrahieren, so wird dies detektiert und das System löscht die Daten (vgl. Shielding).

Um auch Angriffe verhindern zu können, die im ausgeschalteten Zustand des Systems durchgeführt werden, sind die sensiblen Daten mit Hilfe eines kryptografischen Schlüssels, generiert aus den – fertigungstoleranzbedingten – einzigartigen Folieneigenschaften, verschlüsselt im System abgelegt (vgl. PUF). Die kritischen Inhalte werden erst dann decodiert, wenn das System startet oder bereits in Betrieb ist. Das Auslesen der verschlüssel-

Abbildung 1 | Ungeschützte Platine

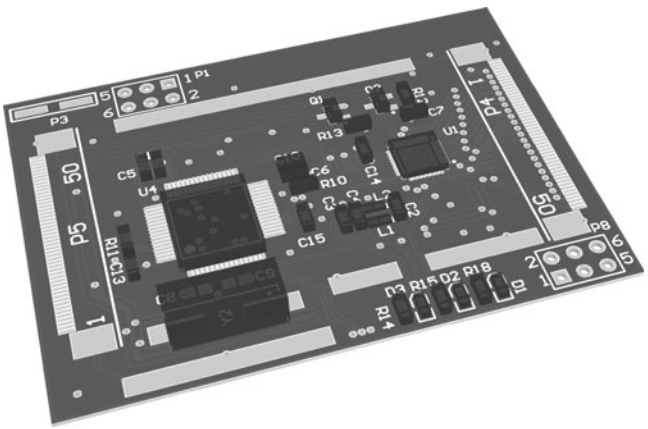


Abbildung 2 | Durch Folie geschützte Platine

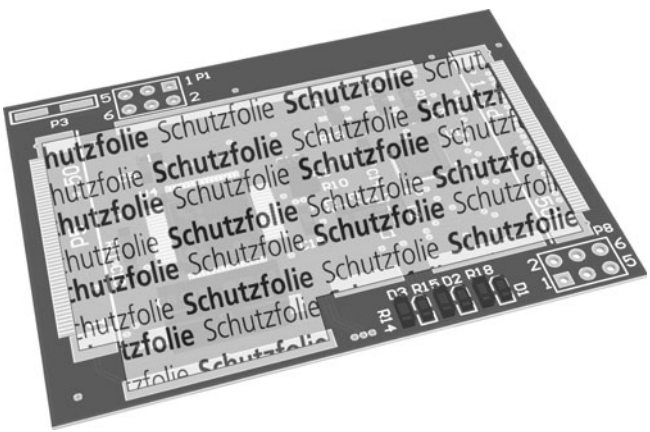
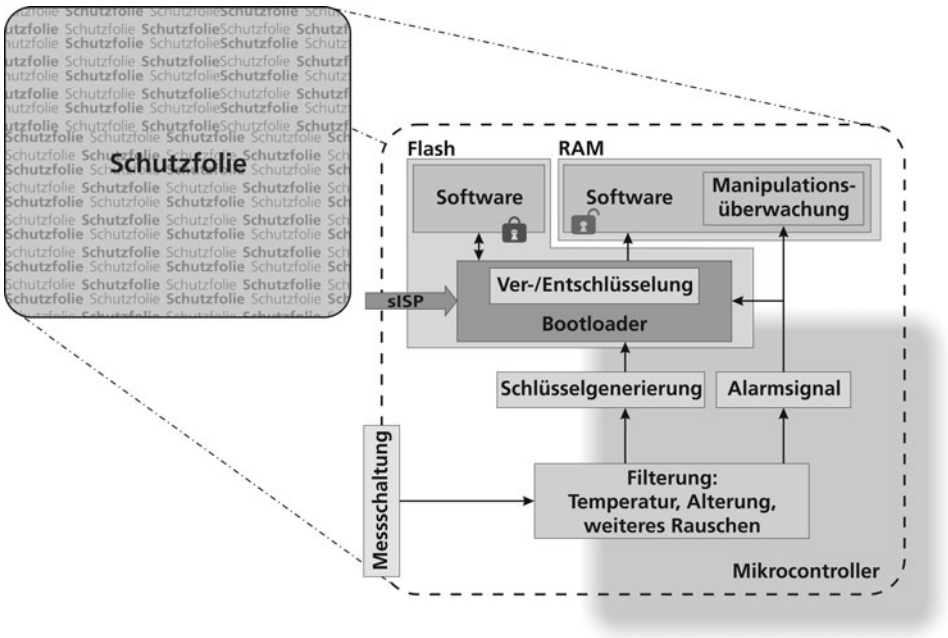


Abbildung 3 | Blockdiagramm des Schutzsystems



ten Daten im ausgeschalteten Zustand ist für einen Produktfälscher dahingehend nutzlos, da die zur Entschlüsselung notwendigen Informationen aus der Folie durch deren Entfernung oder Veränderung der zerstört werden.

4.3 Absicherung aller Betriebsmodi durch Kombination von Schutzmaßnahmen

Um für jeden Betriebszustand (System ist „Aus/Bootet/An") einen ausreichenden Schutz der Software zu gewährleisten, werden im PEP-Konzept drei Schutzmaßnahmen miteinander kombiniert:

- Kritische Bereiche werden mit einer elektronisch auswertbaren Folie abgedeckt und der Folienzustand kontinuierlich überwacht (Shielding)
- Anstelle einer festen Speicherung des kryptografischen Schlüssels wird dieser stets aus den einzigartigen Folieneigenschaften dynamisch abgeleitet (PUF)
- Die Verschlüsselung der Software dient zum Schutz der sensiblen Daten, wenn das System ausgeschaltet ist

Für die initiale Installation der funktionalen Software – oder später auch für Updates – sorgt ein Bootloader, welcher vor dem Aufbringen der Folie in das System eingespielt wird. Nachdem die Folie aufgebracht wurde, kann der Bootloader die Software über eine so genannte secure-In-System-Programming (sISP) Schnittstelle entgegen nehmen (siehe Abbildung 3). Diese Schnittstelle bildet die Verbindung zwischen den unter der Folie befindlichen Bausteinen und der Außenwelt. Die Speicherbereiche, in denen der Bootloader abgelegt ist, sind über diese Schnittstelle allerdings nicht erreichbar. Sie ist zudem so realisiert, dass erst nach einer erfolgreichen Authentisierung bestimmte Speicherbereiche zwar beschrieben, aber nicht ausgelesen werden können.

Beim Starten des Systems (System „Bootet") werden die Eigenschaften der Folie (kapazitiv, induktiv, resistiv) von der ebenfalls unter der Folie befindlichen Messschaltung aufgenommen, Messfehler herausgefiltert und ein kryptografischer Schlüssel (Folien-Schlüssel) abgeleitet. Der Bootloader empfängt nun die funktionale Software, verschlüsselt diese mit dem Folien-Schlüssel und legt sie dauerhaft im Flash-Speicher ab (zum Schutz der Software im Systemzustand „Aus").

Der Bootloader ist im normalen Betrieb zusätzlich dafür zuständig, die verschlüsselte Software aus dem Flash-Speicher zu laden und in den flüchtigen RAM zu entschlüsseln, von welchem sie dann ausgeführt werden kann.

Ein Angreifer könnte nun versuchen, den Bootloader dahingehend zu manipulieren, dass dieser den Folien-Schlüssel oder die entschlüsselte Software über die sISP-Schnittstelle ausgibt. Dazu müsste der Bootloader allerdings vor seiner Installation manipuliert werden. Dies kann durch eine vertrauenswürdige Installationsumgebung vermieden werden. Herstellerseitige Änderung am Bootloader können lediglich durch eine autorisierte Fachkraft mit Zugriff auf die Originalsoftware durchgeführt werden, indem die Folie komplett entfernt, der Bootloader aktualisiert, eine neue Folie aufgebracht und die funktionale Software wieder neu einspielt wird.

Eine weitere Möglichkeit für einen Angreifer wäre nun, die im RAM entschlüsselt vorliegende Software im laufenden Betrieb (System „An") auszulesen, indem er die Folie entfernt und die darunter befindlichen, ungesicherten Programmierschnittstellen der Bausteine kontaktiert. Hierzu ist in der Schutzschaltung unter der Folie eine Laufzeit-Manipulationsüberwachung reali-

Tabelle 1 | Bedrohungen und Wirkungen der Schutzmaßnahmen im Vergleich

Analysemethoden der Produktfälscher			
Schutzmaßnahmen	Identifizieren der Bauteilen	Auslesen/ Analysieren der Software im aktiven Betrieb	Auslesen/Analysieren der Software im ausgeschalteten Zustand
Schutzsiegel	wirkungslos	wirkungslos	wirkungslos
Markierungstechniken	wirkungslos	wirkungslos	wirkungslos
Laserradieren	erschwert	wirkungslos	wirkungslos
Vergussmasse	erschwert	wirkungslos	wirkungslos
Shielding	erschwert	verhindert	wirkungslos
TPM	wirkungslos	erschwert	verhindert
Secure-Elements	wirkungslos	erschwert	verhindert
Dongle-Lösungen	wirkungslos	erschwert	verhindert
PUF	wirkungslos	wirkungslos	verhindert
PEP	erschwert	verhindert	verhindert

siert (siehe Shielding), welche fortlaufend den Zustand der Folie (deren initial während des Bootens erfassten Impedanzen) überwacht und bei Abweichungen einen entsprechenden Alarm auslöst. Dieser löscht den RAM und stoppt das System. Gegebenenfalls kann auch eine Nachricht an den Originalhersteller gesendet werden, um ihn über die versuchte Manipulation zu informieren.

Die RAM-Technologie selbst sorgt zudem dafür, dass die entschlüsselte Software nicht nur bei einer detektierten Manipulation, sondern auch beim Abschalten der Versorgungsspannung gelöscht wird.

5 Zusammenfassung

Eingebettete Systeme lassen sich vor Manipulationen und Nachbauten geeignet schützen. Dafür ist es allerdings notwendig, die Schutzziele eindeutig zu identifizieren, die entsprechenden Schutzmaßnahmen auszuwählen und gezielt anzuwenden.

Abhängig vom Wert des zu schützenden Produktes und den durch eine Kopie potentiell entstehenden finanziellen Schaden oder Image-Verlust für das Unternehmen, können entsprechend angepasste Schutzmaßnahmen gewählt werden. Es gilt für den Originalhersteller dabei stets einen angemessenen Ausgleich zwischen den Investitionen für eine Schutzmaßnahme und dem daraus entstehenden Schutzniveau – respektive dem Aufwand für einen Fälscher, den jener benötigt, um das System zu analysieren und nachzubauen – zu finden. Tabelle 1 zeigt zusammenfassend eine Gegenüberstellung der vorgestellten Schutzmaßnahmen und der Analysemethoden der Produktfälscher. PEP bietet als einzige der Lösungen einen kombinierten Schutzansatz, welcher das System sowohl im aus- als auch im eingeschalteten Zustand absichert.

Ein ähnliches Sicherheitsniveau lässt sich bspw. mit einer Kombination aus Shielding und TPM realisieren. Hier bleibt allerdings das bereits angesprochene Problem der Shielding-Technik, dass der notwendige Referenzwert im System abgelegt sein muss und daher anfällig für Manipulationen ist. Der Systemzustand „Booten" ist hier gefährdet, denn selbst wenn der Referenzwert für den Schutz im Systemzustand „Aus" verschlüsselt abgelegt sein sollte, muss er beim Booten des Systems zunächst einmal entschlüsselt werden, bevor das Shielding einsetzen kann. Damit entsteht ein kleines Zeitfenster, in dem das System angehalten und der Referenzwert ausgelesen und manipuliert werden kann. Das PEP-Konzept löst diese Problematik, indem es die PUF-Technologie und das Shielding kombiniert bzw. überlappend einsetzt.

Das PEP-Konzept zeigt, dass durch eine geeignete Kombination von Schutzmaßnahmen aus Markt und Forschung neue Sicherheitslösungen entwickelt werden können, die das Sicherheitsniveau elektronischer Produkte anhebt. Damit können sowohl das Know-how der Originalhersteller als auch eventuell mit dem Gerät erfasste persönliche Daten besser geschützt werden.

Aktuell befindet sich das PEP-Konzept im fortgeschrittenen Entwicklungsstatus. Die ersten Demonstratoren für einen *Proof of Concept* existieren bereits und es wird fortlaufend an deren Optimierung hinsichtlich der Verträglichkeit mit äußeren Einflüssen und Korrekturverfahren für Alterungsprozesse geforscht. Das Fraunhofer AISEC plant, zum Jahresende 2014 weitere, optimierte Prototypen vorstellen zu können.

Literatur

[1] R. Torrance, D. James: *The State-of-the-Art in IC Reverse Engineering*. In: C. Clavier, K. Gaj (Hrsg.), CHES 2009: 11th International Workshop Lausanne, September 6-9, 2009, Proceedings, LNCS 5747, Springer-Verlag, Berlin (2009), 363-381

[2] B. Filipovic, O. Schimmel: *Schutz eingebetteter Systeme vor Produktpiraterie – Technologischer Hintergrund und Vorbeugemaßnahmen*. Tech. Rep., Fraunhofer AISEC (2011), http://www.aisec.fraunhofer.de/content/dam/aisec/de/pdf/studien/2011-11-15%20Produktschutz-Studie.pdf

[3] WECO: *Vergusstechnik*. http://download.wecogroup.com/them-en/de/WECO_Vergusstechnik.pdf

[4] Flylogic's Analytical Blog: *ST19XL18P – K5F0A Teardown*. http://www.flylogic.net/blog/?p=289

[5] H. Mac Pherson: *Tamper respondent enclosure* (1999), Patent: US 5,858,500

[6] A. Miglioli, V. Ratti, E. Riva, L. Villa: *Tamper respondent enclosure for an electronic device and electrical assembly utilizing same* (2003), Patent: US 6,512,454 B2

[7] D. S. Farquhar, C. Feger, V. Markovich, K. I. Papathomas, M. D. Poliks, J. M. Shaw, G. Szeparowycz, S. H. Weingart: *Tamper-responding encapsulated enclosure having flexible protective mesh structure* (2005), Patent: US 6,929,900 B2

[8] C. Tarnovsky: *Deconstructing a secure processor*. In: Black Hat DC, 2010

[9] D. Merli: *Attacking and Protecting Ring Oscillator Physical Unclonable Functions and Code-Offset Fuzzy Extractors*. München, Technische Universität München, Dissertation, 2014

[10] M. Hennig, O. Schimmel, P. Zieris, G. Sigl: *Manipulationssensible Kopierschutzfolie*. In: DACH 2013, Nürnberg

[11] R. Pappu. *Physical One-Way Functions*. PhD thesis, Massachusetts Institute of Technology, 2001

[12] D. Merli, G. Sigl: *Physical Unclonable Functions – CMOS-Implementierungen und Hardware-Attacken*, In: DuD 12/2012, 876-880

[13] S. Drimer, S. Murdoch, R. Anderson: *Thinking inside the box: system-level failures of tamper proofing*, In: IEEE Symposium on Security and Privacy 2008, 281-295

[14] P. Isaacs, T. Morris Jr, M.J. Fisher and K. Cuthbert, *Tamper Proof, Tamper Evident Encryption Technology*, In: SMTA Proceedings