Vertrauenswürdige und privatheitsbewahrende eingebettete Systeme basierend auf Physically Unclonable Functions*

Christian Wachsmann
Technische Universität Darmstadt
christian.wachsmann@trust.cased.de

Abstract: Die Dissertation präsentiert neuartige Ansätze zum Design und zur Analyse sicherer und privatheitsbewahrender kryptographischer Verfahren für eingebettete Systeme. Sie stellt insbesondere effiziente Authentifikationsprotokolle basierend auf sogenannten Physically Unclonable Functions (PUFs) vor. PUFs erlauben die Erzeugung eindeutiger und unklonbarer Hardware-Fingerprints, die in kryptographische Protokolle eingebunden werden können. Zur Sicherheitsanalyse PUF-basierter Protokolle sowie zur Evaluation der zugrundeliegenden PUF-Implementierungen werden neuartige formale Modelle und Methoden präsentiert. Die aufgestellten Modelle ermöglichen erstmalig die Realisierung beweisbar sicherer PUF-basierter kryptographischer Verfahren und eine Anbindung an komplexitätstheoretische Annahmen im Sinne der modernen Kryptographie. Die Korrektheit der theoretischen Modelle wurde durch die experimentelle Evaluation von PUF-Implementierungen in ASICs validiert.

1 Einleitung

Eingebettete Systeme werden heute in vielen sicherheitskritischen und datenschutzsensiblen Umgebungen eingesetzt. Das Anwendungsspektrum erstreckt sich von Zugangskontrollsystemen, elektronischen Tickets, Sensoren und sogenannten Wearables bis hin zu Automotive-Anwendungen und kritischen Infrastrukturen. In diesen Anwendungen stellt die Authentifikation und die Attestierung, d. h. die Feststellung der eindeutigen Identität und der Integrität der eingebetteten Systeme, eine fundamentale Sicherheitsanforderung dar. Eingebettete Systeme verfügen jedoch meist nur über begrenzte Ressourcen und bieten keine ausreichende Sicherheit, insbesondere hinsichtlich Hardwareangriffen und dem Datenschutz ihrer Nutzer. Eine grundsätzliche Fragestellung in diesem Kontext ist die sichere Bindung von kryptographischen Verfahren an die zugrundeliegende Hardware.

Problemstellung 1: Die Entwicklung von sicheren und privatheitsbewahrenden Authentifikationsverfahren für eingebettete Systeme ist sehr herausfordernd. Dies spiegelt sich in einer Vielzahl von unterschiedlichen Sicherheitsmodellen und Designansätzen wieder (z. B. [Jue05, ADO06, JW07, PV08, BvLDMT09]). Während die in der Praxis eingesetzten Verfahren meist keinen Schutz der Privatsphäre bieten, erfüllen Lösungen aus der Literatur oft die funktionalen Anforderungen praxistauglicher Systeme nicht. Zudem sind

*Englischer Titel der Dissertation: "Trusted and Privacy-preserving Embedded Systems: Advances in Design, Analysis and Application of Lightweight Privacy-preserving Authentication and Physical Security Primitives"

bestehende Sicherheitsmodelle meist nicht miteinander vergleichbar oder gar inkompatibel. So existieren Authentifikationsprotokolle, die zwar in einem Modell formal als sicher bewiesen werden können, in einem anderen jedoch angreifbar sind [JW07]. Um verlässliche Aussagen über die Sicherheit von Authentifikationsverfahren treffen zu können, ist es daher unerlässlich ein einheitliches und praxisnahes Sicherheitsmodell zur Analyse dieser Verfahren zu etablieren.

In diesem Zusammenhang besteht eine grundsätzliche Fragestellung beim Einsatz kryptographischer Verfahren in der sicheren Speicherung der zugrundeliegenden kryptographischen Schlüssel. Insbesondere kosteneffiziente eingebettete Systeme haben oft keinen sicheren Speicher, wodurch die gespeicherten kryptographischen Schlüssel durch Hardwareangriffe ausgelesen werden können. Ein vielversprechender Ansatz diese Systeme vor Hardwareangriffen zu schützen, sind sogenannte Physically Unclonable Functions (PUFs) [MV10]. PUFs basieren auf den im Rahmen von Fertigungstoleranzen entstehenden physikalischen Unterschieden von Hardwarekomponenten, die zwar mit geringem Aufwand gemessen jedoch praktisch nicht reproduziert werden können. Diese Unterschiede sind für jedes Gerät einzigartig und können als Identifikationsmerkmal verwendet werden, sozusagen als physikalischer Fingerabdruck des Gerätes. Dieser Fingerabdruck kann zur Erzeugung kryptographischer Schlüssel verwendet und in sichere kryptographische Protokolle eingebunden werden. Die Vorteile von PUFs bestehen darin, dass kryptographische Schlüssel nicht gespeichert werden müssen, sondern bei Bedarf aus den physikalischen Eigenschaften der PUF erzeugt werden können. Dies ermöglicht zudem eine sichere Bindung dieser Schlüssel und der Software, die diese Schlüssel verwendet, an die zugrundeliegende Hardware-Plattform.

Problemstellung 2: In der Literatur gibt es bereits einige PUF-basierte Authentifikationsverfahren [TB06, BR07]. Jedoch erfordern viele dieser Verfahren die Verfügbarkeit einer Datenbank mit Referenzwerten zur Verifikation des PUF-Fingerabdrucks. Diese Datenbank kann sehr groß werden, insbesondere da jeder Referenzwert nur für eine einzige Authentifikation verwendet werden darf, da sonst Replay-Angriffe möglich sind. Ein anderer Ansatz nutzt die PUF zur Erzeugung kryptographischer Schlüssel zur Verwendung in Standard-Authentifikationsverfahren und erfordert den Einsatz von Fehlerkorrekturmechanismen, um die Reproduzierbarkeit des Schlüssels zu gewährleisten. Die Implementierungen der zugrundeliegenden Dekodierverfahren sind jedoch oft sehr komplex und für eingebettete Systeme ungeeignet.

Problemstellung 3: Die Sicherheit von PUF-basierten Protokollen fundiert auf physikalischen Annahmen, die bisher noch nicht hinreichend untersucht wurden. So werden in der Literatur oft idealisierte Sicherheitsmodelle für PUFs verwendet, die nicht alle Eigenschaften von PUF-Implementierungen berücksichtigen. Zudem gibt es bisher kein allgemeingültiges formales Sicherheitsmodell für PUF-basierte kryptographische Verfahren.

2 Zusammenfassung der Ergebnisse der Dissertation

Die Dissertation [Wac14] stellt neuartige Ansätze zum Design und zur Analyse sicherer und privatheitsbewahrender kryptographischer Authentifikationsverfahren vor, die für verschiedenartige eingebettete Systeme geeignet sind. Die zugrundeliegende Arbeit leis-

tet einen wesentlichen Beitrag zur Entwicklung eines einheitlichen und praxisnahen Sicherheitsmodells für Authentifikationsverfahren für eingebettete Systeme. Konkret zeigen wir am Beispiel eines der bis dato umfassendsten Sicherheitsmodelle für diese Systeme [PV08], dass subtile Aspekte bei der Modellierung von Hardwareangriffen dazu führen können, dass manche Sicherheits- und Datenschutzziele formal nicht gleichzeitig erreicht werden können. Unsere Ergebnisse bilden die Grundlage von wissenschaftlichen Folgearbeiten zum Design und zur Analyse von privatheitsbewahrenden Authentifikationsverfahren (z. B. [Vau10, HPVP11, DR12]).

Beim Design der von uns vorgestellten Authentifikationsverfahren betrachten wir insbesondere praxisnahe Lösungen basierend auf Physically Unclonable Functions (PUFs), die ohne die sichere Speicherung kryptografischer Schlüssel auskommen und somit den Angriffsvektor reduzieren. Zur Sicherheitsanalyse dieser Verfahren sowie zur Evaluation der zugrundeliegenden PUF-Implementierungen präsentieren wir neuartige formale Werkzeuge: ein Evaluations-Framework für PUF-Implementierungen und ein formales Sicherheitsmodell für PUF-basierte kryptographische Verfahren. Unser Evaluations-Framework erlaubt die einheitliche Analyse verschiedener PUF-Typen und eine präzisere Bewertung der von PUFs generierten Entropie als vorherige Methoden. Basierend auf unserem Evaluations-Framework präsentieren wir eine umfangreiche Analyse der wichtigsten Eigenschaften von Implementierungen unterschiedlicher PUF-Typen (Arbiter, Ring Oszillator, SRAM, Flip-Flop und Latch PUFs) in ASIC. Unsere Ergebnisse erlauben erstmals den direkten und fairen Vergleich der Eigenschaften verschiedener PUF-Typen. Unser formales Sicherheitsmodell ermöglicht erstmalig die Realisierung beweisbar sicherer PUF-basierter kryptographischer Verfahren.

Unsere Ergebnisse geben neue Einsichten in die Nutzung physikalischer Eigenschaften von Hardware zur eindeutigen Authentifikation von eingebetteten Systemen. Unsere formalen und praxisnahen Lösungen adressieren die im Bereich der IT-Sicherheit grundsätzliche Problemstellung der eindeutigen und unklonbaren Bindung von Protokollen an die zugrundeliegende Hardware und haben eine Reihe wissenschaftlicher Arbeiten und Veröffentlichungen (z. B. [Vau10, HPVP11, DR12, SSJM12, SKS12, DGK+12]) auf international renommierten Konferenzen erzeugt und motiviert. In den folgenden Abschnitten stellen wir die wichtigsten Ergebnisse der Dissertation detaillierter vor.

3 Hintergrundinformationen zu Physically Unclonable Functions

Eine Physically Unclonable Function (PUF) ist ein physikalisches System, das in ein physikalisches Objekt (z. B. einen Mikrochip) eingebettet ist [MV10]. Es gibt eine Vielzahl von PUF-Typen, welche auf den verschiedensten physikalischen Merkmalen basieren, darunter optische, magnetische und elektrische Effekte. Für die Integration in eingebettete Systeme am geeignetsten sind PUFs basierend auf elektrischen Effekten. Zu den wichtigsten elektrischen PUFs zählen sogenannte verzögerungsbasierte (delay-based) PUFs und speicherbasierte (memory-based) PUFs. Verzögerungsbasierte PUFs basieren auf Signallaufzeiten in elektronischen Schaltungen (z. B. Arbiter und Ring Oszillator PUF) während speicherbasierte PUFs auf der Instabilität von flüchtigen Speicherzellen, wie SRAM, Flip-Flops und Latches basieren.

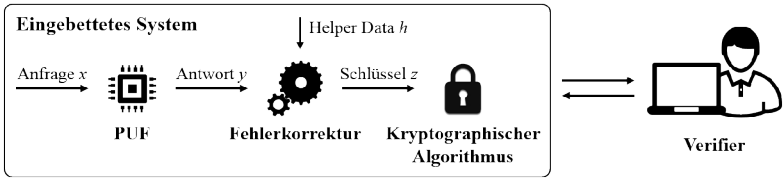

Abbildung 1: Typisches Anwendungsszenario von Physically Unclonable Functions

Wenn eine PUF mit einer Anfrage x (z. B. einem elektrischen Signal) stimuliert wird, antwortet sie mit einer eindeutigen Antwort $y \leftarrow \mathsf{PUF}(x)$, die von den physikalischen Eigenschaften der PUF und x abhängig ist (siehe Abbildung 1). Da die physikalischen Eigenschaften der PUF-Hardware auch von den Betriebsbedingungen (z. B. Schwankungen in der Umgebungstemperatur und der Versorgungsspannung) der PUF-Hardware beeinflusst werden, wird eine PUF auf die selbe Anfrage x immer leicht unterschiedlich antworten.

Die wichtigsten in der Literatur getroffenen Annahmen über PUFs sind *Robustheit*, *physikalische Unklonbarkeit* und *Unberechenbarkeit*. Robustheit bedeutet informell, dass eine mehrfach mit der selben Anfrage x stimulierte PUF immer mit ähnlichen Antworten $y_i = y + e_i$ antwortet, wobei die Unterschiede (z. B. das Hamming-Gewicht von) e_i gering sind. Physikalische Unklonbarkeit bedeutet, dass es praktisch unmöglich ist, zwei PUFs herzustellen, die nicht anhand ihres Anfrage/Antwort-Verhaltens unterschieden werden können. Unberechenbarkeit bezeichnet die Eigenschaft, dass es praktisch unmöglich ist, die Antwort y einer PUF auf eine zufällig gewählte Anfrage x vorherzusagen.

PUFs können zur Erzeugung von kryptographischen Schlüsseln z verwendet und in kryptographische Protokolle eingebunden werden. Um die Reproduzierbarkeit von z zu gewährleisten, werden PUFs meist in Kombination mit Fehlerkorrekturverfahren (z. B. Secure Sketches [DRS04]) eingesetzt (siehe Abbildung 1). Üblicherweise wird vor dem Einsatz der PUF für jede Anfrage x (oder für eine Teilmenge von Anfragen) eine Antwort y als Referenzwert ermittelt. Zusätzlich wird mittels des Fehlerkorrekturverfahrens eine sogenannte Helper Data h für y erzeugt. Später, wenn die PUF auf die Anfrage x mit $y_i = y + e_i$ antwortet, kann das Fehlerkorrekturverfahren mit Hilfe von h den Fehler e_i innerhalb der Grenzen des zugrundeliegenden Dekodierungsverfahrens korrigieren und den Referenzwert y reproduzieren. Hierbei ist zu beachten, dass h zwar partielle Informationen über y preisgibt, jedoch aber nicht geheim gehalten werden muss. Um aus y einen kryptographischen Schlüssel z zu erzeugen, muss die in y enthaltene Entropie (z. B. mittels kryptographischer Hash-Funktionen) extrahiert werden. Hierbei muss der durch das Veröffentlichen von h verursachte Entropieverlust in y berücksichtigt werden. Typischerweise wird der Partei (*Verifier*), die später ein kryptographisches Protokoll mit dem eingebetteten System ausführt, bei der Systeminitialisierung der aus y erzeugte Schlüssel z über einen sicheren Kanal mitgeteilt. Die PUF-Antwort y und Helper Data h können im eingebetteten System abgespeichert werden und müssen nicht geheim gehalten werden.

4 Analyse der Eigenschaften von PUF-Implementierungen

Die Robustheit und die Unberechenbarkeit Eigenschaften von PUFs sind essentiell für deren Integration in sichere und zuverlässige kryptographische Verfahren. In der Literatur finden sich unterschiedliche Ansätze zur Analyse der Eigenschaften von PUF-Implementierungen [TvS+05, HBF09]. Jedoch sind die Ergebnisse dieser Analysen aufgrund der unterschiedlichen Methoden und Testbedingungen nur schwer vergleichbar. Darüberhinaus sind sie nicht konform zu etablierten Sicherheitsmodellen in der modernen Kryptographie.

Wir stellen ein neuartiges Evaluations-Framework vor, das erstmals die einheitliche Analyse der wichtigsten Eigenschaften von Implementierungen unterschiedlicher PUF-Typen ermöglicht. Unser Framework erlaubt zudem eine präzisere Bewertung der Güte von PUFs, insbesondere der generierten Entropie, als vorherige Ansätze.

Unsere Analyse verwendet die Shannon Entropie, die üblicherweise in der Kryptographie betrachtet wird. Insbesondere interessieren wir uns für die minimale Entropie eines einzigen PUF-Antwort-Bits unter der Bedingung, dass alle anderen Bits der Antwort bekannt sind. Dies ermöglicht die Abschätzung einer informationstheoretischen oberen Schranke für die Wahrscheinlichkeit, dass ein starker Angreifer ein einzelnes Bit der PUF-Antwort vorhersagen kann, selbst wenn er alle anderen Bits der Antwort kennt. Formal betrachten wir die bedingte Min-Entropie

$$\mathbf{H}_\infty(Y|W) = -\log_2\big(\max_{x\in X}\{\Pr\big[Y(x)|W(x)\big]\}\big). \tag{1}$$

Hierbei bezeichnet X die Menge aller PUF-Anfragen, $Y(x)$ die Zufallsvariable für das Antwort-Bit y bezüglich einer Anfrage x und $W(x)$ die Zufallsvariable für die Menge aller Antwort-Bits ohne y, also $W(x) = \big\{y'|y' \leftarrow \mathsf{PUF}(x') \wedge x' \in X \setminus \{x\}\big\}$. Die Berechnung dieser Entropie ist praktisch nur schwer durchführbar, da deren Komplexität exponentiell mit der Größe von X und der Größe der Menge aller PUF-Antworten wächst. Um die Entropie dennoch abschätzen zu können, treffen wir die im Folgenden erläuterten Annahmen über die physikalischen Eigenschaften der betrachteten PUFs.

Alle bis dato bekannten elektrischen PUFs bestehen aus einzelnen Elementen (z. B. Speicherzellen), die sich an verschiedenen Positionen im Chip befinden. Unter der Annahme, dass sich weit voneinander entfernte PUF-Elemente gegenseitig kaum beeinflussen und daher keine statistischen Abhängigkeiten aufweisen [HBF09, MV10], lässt sich die bei der Entropieberechnung zu berücksichtigende Menge an PUF-Antwort-Bits signifikant reduzieren. Insbesondere betrachten wir statt der Menge der Antwort-Bits $W(x)$ die deutlich kleinere Menge aller Antwort-Bits $W'(x)$, die von PUF-Elementen in direkter physischer Nähe zum PUF-Element des Antwort-Bits $Y(x)$ erzeugt wurden. Dies ermöglicht eine effiziente und dennoch präzise Abschätzung von Gleichung 1.

Basierend auf unserem Evaluations-Framework haben wir unterschiedliche Implementierungen verschiedener PUF-Typen in ASICs analysiert. Unsere Evaluation basiert auf Daten aus 96 ASICs, die in 65 nm CMOS Technologie mit Industriepartnern hergestellt wurden. Jeder dieser ASICs enthält mehrere Implementierungen unterschiedlicher PUF-Typen, darunter Ring Oszillator, Arbiter, SRAM, Flip-Flop und Latch PUFs.

Unsere Evaluationsergebnisse zeigen, dass alle untersuchten PUFs selbst unter verschie-

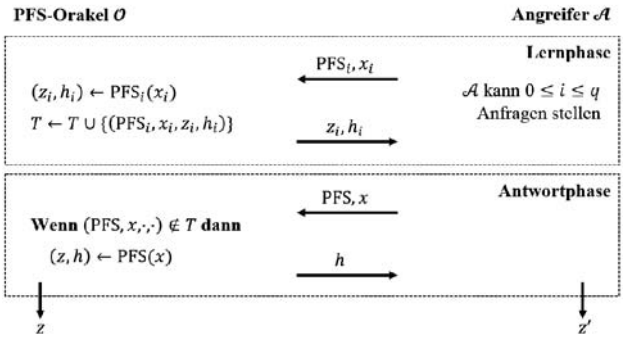

Abbildung 2: Sicherheitsexperiment $\mathbf{Exp}_{\mathcal{A}}^{\text{w-uprd}}(q)$ für Unberechenbarkeit

denen Betriebsbedingungen und hinsichtlich der Materialermüdung ausreichend robust für praktische Anwendungen sind. Jedoch ist die Entropie der PUF-Antworten stark vom PUF-Typ und den Einsatzbedingungen der PUF-Hardware abhängig. Insbesondere die von Arbiter PUFs generierte Entropie ist sehr niedrig, während die Entropie der Flip-Flop und Latch PUF-Antworten durch Temperaturschwankungen beeinflusst wird. Die von Ring Oszillator und SRAM PUFs generierte Entropie ist nahezu ideal und ändert sich bei unterschiedlichen Betriebsbedingungen kaum.

5 Sicherheitsmodell für PUF-basierte kryptographische Verfahren

Wir präsentieren ein formales Sicherheitsmodell für PUFs, welches erstmals eine fundierte Analyse von PUF-basierten kryptographischen Verfahren ermöglicht. Das Modell berücksichtigt die wichtigsten Eigenschaften von PUFs: Robustheit, physikalische Unklonbarkeit und Unberechenbarkeit.

Unser Modell umfasst alle Komponenten, die zum Einsatz von PUFs in kryptographischen Verfahren erforderlich sind. Diese Komponenten fassen wir als *Physical Function System* (PFS) zusammen, das im Wesentlichen einer mit einer PUF ausgestatteten Hardware-Plattform entspricht. Hierbei betrachten wir *physikalische Funktionen* (PF) im Allgemeinen wobei die physikalische Unklonbarkeit nur ein mögliches Merkmal einer PF darstellt. Eine PF besteht aus einer physikalischen Komponente C (z. B. einem Schaltkreis) und einem Evaluationsprozess, der als Schnittstelle zu C dient. Die physikalische Komponente C wird durch einen Herstellungsprozess erzeugt, der gewissen Fertigungstoleranzen unterliegt aus denen die Einzigartigkeit von C resultiert. Um durch Schwankungen in den Betriebsbedingungen der PF-Hardware hervorgerufene Einflüsse auf die PF-Antworten y zu kompensieren, werden PFs meist in Kombination mit einem Extraktor (z. B. einem Fehlerkorrekturverfahren) verwendet [DRS04, MV10]. Der Extraktor kann in zwei Modi betrieben werden: Erzeugung und Rekonstruktion. Im Erzeugung-Modus erzeugt der Extraktor eine Helper Data h und einen Schlüssel z aus y. Im Rekonstruktion-Modus rekonstruiert der Extraktor z aus h und $y_i = y + e_i$.

Basierend auf unserem Systemmodell formalisieren wir Robustheit, physikalische Un-

klonbarkeit und Unberechenbarkeit in Anlehnung an etablierte kryptographische Modelle. Im Folgenden beschreiben wir beispielhaft die Formalisierung von Unberechenbarkeit, die auf dem in Abbildung 2 dargestelltem Sicherheitsexperiment $\mathbf{Exp}_{\mathcal{A}}^{\text{w-uprd}}(q)$ basiert. In diesem Experiment darf ein Angreifer $\mathcal{A}$ in einer Lernphase zunächst bis zu q Anfragen x_i an verschiedene PFS_i stellen und erhält die jeweiligen Ausgaben (z_i, h_i). Die in der Lernphase gewonnene Information kann $\mathcal{A}$ in der Antwortphase des Experiments nutzen, um eine ihm bisher *unbekannte* Ausgabe z eines PFS vorherzusagen. Ein PFS ist unberechenbar, wenn $\mathcal{A}$ dies nur mit geringer Wahrscheinlichkeit gelingt, formal:

Definition 1 (Unberechenbarkeit)**.** *Sei $T = \emptyset$, $\lambda, q \in \mathbb{N}$ mit $q \geq 0$ und $\mathbf{Exp}_{\mathcal{A}}^{\text{w-uprd}}(q)$ das in Abbildung 2 dargestellte Sicherheitsexperiment. Weiterhin bezeichne ρ die Robustheit des PFS. Ein PFS ist* (λ, q)-unberechenbar*, wenn gilt:*

$$\Pr\left[z = z' | (z, z') \leftarrow \mathbf{Exp}_{\mathcal{A}}^{\text{w-uprd}}(q)\right] \leq \lambda \cdot \rho$$

Die Definition ist ähnlich zur Definition von kryptographischen Pseudorandom Functions (PRFs). Jedoch gibt es subtile aber entscheidende Unterschiede, die von den PUF-Modellen in der Literatur nicht betrachtet wurden. Zum einen muss die Helper Data h einbezogen werden, die $\mathcal{A}$ Informationen über z preisgeben könnte. Zum anderen kann $\mathcal{A}$ im Gegensatz zur Sicherheitsdefinition von PRFs mit mehreren verschiedenen PFS_i interagieren, was die Möglichkeit berücksichtigt, dass $\mathcal{A}$ basierend auf den Ausgaben eines PFS Aussagen über die Ausgaben eines anderen PFS treffen könnte.

Unser Sicherheitsmodell dient als Grundlage für Folgearbeiten zum Design von PUFs und PUF-basierter kryptographischer Verfahren (z. B. [SSJM12, SKS12, DGK+12]).

6 Physikalisch-kryptographische Verfahren für eingebettete Systeme

Bestehende PUF-basierte kryptographische Verfahren erfordern entweder eine große Datenbank mit Referenzwerten zur Verifikation der PUF-Ausgaben oder die Implementierung von komplexen Dekodierverfahren. Beides ist für praktische Anwendungen eingebetteter Systeme ungeeignet. Wir stellen ein PUF-basiertes Authentifikationsverfahren für eingebettete Systeme vor, das ressourcenschonend implementierbar ist und keine große Referenzdatenbank benötigt.

Das Protokoll ist in Abbildung 3 dargestellt. Bei der Initialisierung des Verfahrens wird mindestens eine Anfrage x und die dazugehörige Antwort y der Physical Function (PF) des eingebetteten Systems $\mathcal{P}$ in der Datenbank D des Verifiers $\mathcal{V}$ gespeichert. $\mathcal{V}$ startet das Protokoll indem er eine zufällig gewählte PF-Anfrage x und einen zufällig gewählten Wert c mit n-Bit Länge an $\mathcal{P}$ schickt. Daraufhin ermittelt $\mathcal{P}$ die Antwort y seiner PF, erzeugt daraus eine Helper Data h und einen Schlüssel z und schickt h zusammen mit dem mittels einer kryptographischen Hash-Funktion berechneten Wert r an $\mathcal{V}$. Schließlich rekonstruiert $\mathcal{V}$ den von $\mathcal{P}$ verwendeten Schlüssel z, berechnet die Hash-Funktion und akzeptiert $\mathcal{P}$ nur dann, wenn das Ergebnis mit dem von $\mathcal{P}$ erhaltenem r übereinstimmt.

In unserem Protokoll wird der Extraktor anders als in der Literatur üblich eingesetzt, um auf der Seite von $\mathcal{V}$ den von $\mathcal{P}$ verwendeten Schlüssel z zu rekonstruieren. Standard-

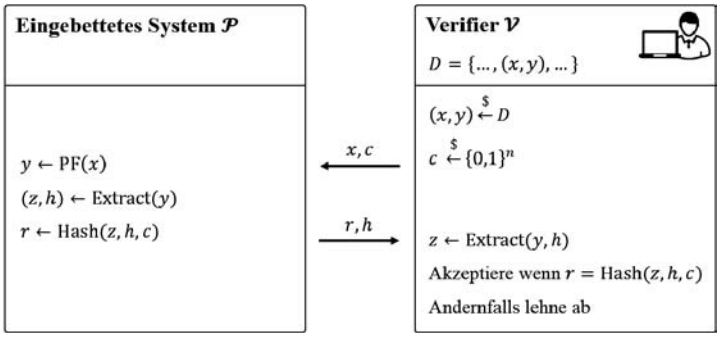

Abbildung 3: PUF-basiertes Authentifikationsprotokoll

Implementierungen des Extraktors sind in diesem Szenario nicht anwendbar, da ein Angreifer mehrere Helper Data zu PUF-Antworten auf die selbe Anfrage x erhalten und somit genug Informationen sammeln kann, um den Schlüssel z zu rekonstruieren. Daher erfordert dieser Ansatz die Verwendung spezieller Extractoren [Boy04], die auch in diesem Szenario als sicher gelten.

Wir zeigen die Sicherheit unseres Protokolls formal nach dem in der Kryptographie üblichen komplexitätstheoretischen Prinzip der Reduktion. Konkret zeigen wir, dass ein Angreifer $\mathcal{A}$, der eine Nachricht (r, h) erzeugen kann, die der Verifier $\mathcal{V}$ akzeptiert, entweder die Sicherheit des Extraktors, der Hash-Funktion oder die Unberechenbarkeit der PUF verletzen kann. Basierend auf der Annahme, dass die verwendete Hash-Funktion und der Extraktor ihre Sicherheitsansprüche erfüllen und auf unseren Evaluationsergebnissen, die zeigen, dass die Antworten geeigneter PUF-Implementierungen eine nahezu ideale Entropie erreichen und somit praktisch nicht berechenbar sind, folgern wir, dass ein solcher Angreifer in der Praxis nicht existiert.

7 Zusammenfassung und Ausblick

Wir präsentieren sichere, effiziente und praxisnahe PUF-basierte kryptographische Protokolle und formale Werkzeuge, die eine fundierte Sicherheitsanalyse dieser Verfahren ermöglichen. Konkret stellen wir ein neuartiges formales Sicherheitsmodell für PUF-basierte kryptographische Verfahren und ein Evaluations-Framework für PUF-Implementierungen vor. Unsere Ergebnisse sind die Grundlage für mehrere wissenschaftliche Folgearbeiten zur Erfassung der Eigenschaften neuartiger PUF-Typen und beim Design PUF-basierter kryptographischer Verfahren.

Unsere Arbeit liefert Anknüpfungspunkte für aktuelle Forschung. Während bisherige Implementierungen verzögerungsbasierter PUFs meist als zusätzliche Schaltung implementiert werden müssen, stellen wir ein neuartiges PUF-Design vor, das es ermöglicht vorhandene Schaltungen in eingebetteten Systemen neben ihrer eigentlichen Funktion auch als PUF zu verwenden [KKSW14]. Ein weiterer Anknüpfungspunkt ist die Erweiterung unseres PUF-Evaluations-Frameworks hinsichtlich Hardwareangriffen. In diesem Kontext

zeigen wir, dass speicherbasierte PUFs unter bestimmten Bedingungen durch die Kombination von Hardwareangriffen und Kryptanalyse angegriffen werden können und präsentieren Gegenmaßnahmen [OSW13].

Eine weitere grundsätzliche Fragestellung neben der Authentifikation von eingebetteten Systemen ist die Verifizierung der Integrität von entfernten Rechnerplattformen (Attestierung). In diesem Zusammenhang analysieren wir Attestierungsprotokolle für eingebettete Systeme mit stark begrenzten Ressourcen [ASSW13]. Durch die Integration von PUFs binden wir diese Protokolle an die zugrundeliegende Hardware, wodurch bestimmte Angriffe verhindert werden [SSW11].

Literatur

[ADO06] G. Avoine, E. Dysli und P. Oechslin. Reducing Time Complexity in RFID Systems. In *Selected Areas in Cryptography (SAC)*, Jgg. 3897 of *LNCS*, Seiten 291–306. Springer, 2006.

[ASSW13] F. Armknecht, A.-R. Sadeghi, S. Schulz und C. Wachsmann. A Security Framework for the Analysis and Design of Software Attestation. In *ACM Conference on Computer and Communications Security (CCS)*, Seiten 1–12. ACM, 2013.

[Boy04] X. Boyen. Reusable Cryptographic Fuzzy Extractors. In *ACM Conference on Computer and Communications Security (CCS)*, Seiten 82–91. ACM, 2004.

[BR07] L. Bolotnyy und G. Robins. Physically Unclonable Function-based Security and Privacy in RFID Systems. In *Conference on Pervasive Computing and Communications (PerCom)*, Seiten 211–220. IEEE, 2007.

[BvLDMT09] M. Burmester, T. van Le, B. De Medeiros und G. Tsudik. Universally Composable RFID Identification and Authentication Protocols. *ACM Transactions on Information and Systems Security*, 12(4), 2009.

[DGK+12] F. Durvaux, B. Gérard, S. Kerckhof, F. Koeune und F. Standaert. Intellectual Property Protection for Integrated Systems Using Soft Physical Hash Functions. In *Information Security Applications*, Jgg. 7690 of *LNCS*, Seiten 208–225. Springer, 2012.

[DR12] T. Deursen und S. Radomirović. Insider Attacks and Privacy of RFID Protocols. In *European Conference on Public Key Infrastructures, Services and Applications (EuroPKI)*, Jgg. 7163 of *LNCS*, Seiten 91–105. Springer, 2012.

[DRS04] Y. Dodis, L. Reyzin und A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In *Advances in Cryptology (EUROCRYPT)*, Jgg. 3027 of *LNCS*, Seiten 523–540. Springer, 2004.

[HBF09] D. Holcomb, W. P. Burleson und K. Fu. Power-up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, 2009.

[HPVP11] J. Hermans, A. Pashalidis, F. Vercauteren und B. Preneel. A New RFID Privacy Model. In *European Symposium on Research in Computer Security (ESORICS)*, Jgg. 6879 of *LNCS*, Seiten 568–587. Springer, 2011.

[Jue05] A. Juels. Minimalist Cryptography for Low-cost RFID Tags (Extended Abstract). In *Security in Communication Networks (SCN)*, Jgg. 3352 of *LNCS*, Seiten 149–164. Springer, 2005.

[JW07] A. Juels und S. A. Weis. Defining Strong Privacy for RFID. In *Conference on Pervasive Computing and Communications (PerCom)*, Seiten 342–347. IEEE, 2007.

[KKSW14] J. Kong, F. Koushanfar, A.-R. Sadeghi und C. Wachsmann. PUFatt: Embedded Platform Attestation Based on Novel Processor-Based PUFs. In *Design Automation Conference (DAC)*. 2014.

[MV10] R. Maes und I. Verbauwhede. Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions. In *Towards Hardware-Intrinsic Security*, Seiten 3–37. Springer, 2010.

[OSW13] Y. Oren, A.-R. Sadeghi und C. Wachsmann. On the Effectiveness of the Remanence Decay Side-Channel to Clone Memory-based PUFs. In *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, Jgg. 8086 of *LNCS*, Seiten 107–125. Springer, 2013.

[PV08] R. I. Paise und S. Vaudenay. Mutual Authentication in RFID: Security and Privacy. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Seiten 292–299. ACM, 2008.

[SKS12] S. Shariati, F. Koeune und F. Standaert. Security Analysis of Image-based PUFs for Anti-counterfeiting. In *Communications and Multimedia Security*, Jgg. 7394 of *LNCS*, Seiten 26–38. Springer, 2012.

[SSJM12] S. Shariati, F.-X. Standaert, L. Jacques und B. Macq. Analysis and Experimental Evaluation of Image-based PUFs. 2(3):189–206, 2012.

[SSW11] S. Schulz, A.-R. Sadeghi und C. Wachsmann. Short Paper: Lightweight Remote Attestation Using Physical Functions. In *ACM Conference on Wireless Network Security (WiSec)*, Seiten 109–114. ACM, 2011.

[TB06] P. Tuyls und L. Batina. RFID-tags for Anti-counterfeiting. In *Topics in Cryptology (CT-RSA)*, Jgg. 3860 of *LNCS*, Seiten 115–131. Springer, 2006.

[TvS+05] P. Tuyls, B. Škorić, S. Stallinga, A. H. M. Akkermans und W. Ophey. Information-Theoretic Security Analysis of Physical Uncloneable Functions. In *Financial Cryptography and Data Security (FC)*, Jgg. 3570 of *LNCS*, Seite 578. Springer, 2005.

[Vau10] S. Vaudenay. Privacy Models for RFID Schemes. In *Radio Frequency Identification: Security and Privacy Issues (RFIDSec)*, Jgg. 6370 of *LNCS*, Seite 65. Springer, 2010.

[Wac14] C. Wachsmann. *Trusted and Privacy-preserving Embedded Systems: Advances in Design, Analysis and Application of Lightweight Privacy-preserving Authentication and Physical Security Primitives*. Dissertation, TU Darmstadt, 2014.

Christian Wachsmann hat im September 2013 seine Promotion an der TU Darmstadt mit Auszeichnung abgeschlossen. Zurzeit ist er am Intel Collaborative Research Institute for Secure Computing an der TU Darmstadt beschäftigt. Der Schwerpunkt seiner Arbeit liegt in der Konzeption, Entwicklung, formalen Modellierung und Sicherheitsanalyse von Sicherheitsarchitekturen und kryptographischen Protokollen zur Überprüfung der Softwareintegrität (Attestierung) von eingebetteten Systemen und hardwarebasierten Sicherheitsmechanismen zur Erkennung von Produktfälschungen. Christian Wachsmann ist Autor von über 30 wissenschaftlichen Veröffentlichungen in international renommierten Fachzeitschriften und IT-Sicherheitskonferenzen.