SHIVA: Sichere Hardware in der Informationsverarbeitung

Kochte, Michael A.; Sauer, Matthias; Raiola, Pascal; Becker, Bernd; Wunderlich, Hans-Joachim

Proceedings of the ITG/GI/GMM edaWorkshop 2016 Hannover, Germany, 11-12 May 2016

url: http://www.book-on-demand.de/shop/14818

Abstract: Das Projekt "SHIVA: Sichere Hardware in der Informationsverarbeitung" ist Teil des Forschungsprogramms "IKT-Sicherheit für weltweit vernetzte vertrauenswürdige Infrastrukturen" der Baden-Württemberg Stiftung. Ziel des Projekts sind die Erforschung von Entwurfs- und Verifikationsmethoden zur Steigerung der Sicherheit mikroelektronischer Hardware, beispielsweise aus der Automobilelektronik, der Medizintechnik oder auch der Fertigungstechnik. Es soll damit die missbräuchliche Verwendung nichtfunktionaler Hardware-Infrastruktur zur Beobachtung interner sensibler Daten, verwendeter Verfahren und Prozesse sowie zu Angriffen auf das geistige Eigentum an der Hardware ausgeschlossen werden. Das Projekt ist eine Kooperation des Instituts für Technische Informatik (ITI) der Universität Stuttgart und des Lehrstuhls für Rechnerarchitektur der Universität Freiburg. Dieser Beitrag stellt die Projektziele und erste Forschungsergebnisse vor.

Preprint

SHIVA: Sichere Hardware in der Informationsverarbeitung

Formaler Nachweis komplexer Sicherheitseigenschaften in rekonfigurierbarer Infrastruktur

Kochte, Michael, ITI, Universität Stuttgart, 70569 Stuttgart, Deutschland
Sauer, Matthias, Universität Freiburg, 79110 Freiburg, Deutschland
Raiola, Pascal, Universität Freiburg, 79110 Freiburg, Deutschland
Becker, Bernd, Universität Freiburg, 79110 Freiburg, Deutschland
Wunderlich, Hans-Joachim, ITI, Universität Stuttgart, 70569 Stuttgart, Deutschland

***Zusammenfassung*—Das Projekt „SHIVA: Sichere Hardware in der Informationsverarbeitung“ ist Teil des Forschungsprogramms „IKT-Sicherheit für weltweit vernetzte vertrauenswürdige Infrastrukturen“ der Baden-Württemberg Stiftung. Ziel des Projekts sind die Erforschung von Entwurfs- und Verifikationsmethoden zur Steigerung der Sicherheit mikroelektronischer Hardware, beispielsweise aus der Automobilelektronik, der Medizintechnik oder auch der Fertigungstechnik. Es soll damit die missbräuchliche Verwendung nicht-funktionaler Hardware-Infrastruktur zur Beobachtung interner sensibler Daten, verwendeter Verfahren und Prozesse sowie zu Angriffen auf das geistige Eigentum an der Hardware ausgeschlossen werden.**

Das Projekt ist eine Kooperation des Instituts für Technische Informatik (ITI) der Universität Stuttgart und des Lehrstuhls für Rechnerarchitektur der Universität Freiburg. Dieser Beitrag stellt die Projektziele und erste Forschungsergebnisse vor.

I. Motivation

Sichere Informations- und Kommunikationstechnologien erfordern integrierte und aufeinander abgestimmte Schutzmaßnahmen auf allen Ebenen, beginnend von der Systemarchitektur über die Anwendungen und die Software, die Hardwarearchitektur, die Bausteinebene bis hin zur elektrischen Ebene. Isolierte Schutzmaßnahmen auf einer oder mehreren Ebenen werden entwertet, wenn Angriffe über andere Wege möglich sind.

Eine besondere Rolle spielt hier die Hardware eines sicheren IKT-Systems, da sie neben funktionalen Angriffsmöglichkeiten, die auch die Software bietet, zahlreiche nicht-funktionale Angriffskanäle aufweist. Zu diesen gehören beispielsweise elektromagnetische Abstrahlung, Stromverbrauch und insbesondere die *nicht-funktionale Infrastruktur*. Diese integrierte Chip-Infrastruktur ist notwendig, um mittels kontrolliertem Zugriff auf die Test-, Diagnose- und Wartbarkeitsschnittstellen der Hardware während der Fertigung als auch im Feld einen wirtschaftlichen und zuverlässigen Betrieb zu gewährleisten. Allerdings eröffnet diese Infrastruktur zahlreiche Angriffsmöglichkeiten und kann das System verwundbar machen.

Einer ganz besonderen Gefahr sind die sogenannten „Cyber Physical Systems“ (CPS) ausgesetzt, zu denen sicherheitskritische Systeme im Bereich der Automobilelektronik, der Medizintechnik oder auch der Fertigungstechnik (Industrie 4.0) gehören, da sie einem potentiellen Angreifer auch unmittelbar physisch zugänglich sein können.

Das Projekt „Trustworthy Embedded Networked Systems“ (TENSE) der KU Leuven analysiert das Zusammenspiel von Sicherheitsfeatures in Software und Hardware. In den USA wurde die Erkennung gefälschter Hardware und Hintertüren in dem DARPA Projekt TRUST „Trusted Integrated Circuits“ betrachtet. Das „Center for Hardware Assurance, Security, and Engineering“ (CHASE) der Universität Connecticut erforscht Entwurfsmethoden für sichere Hardware. Das vom BMBF geförderte nationale Referenzprojekt IUNO *IT-Sicherheit in Industrie 4.0* entwickelt Sicherheitslösungen für vernetzte industrielle Maschinen- und Steuerungsanlagen. Das Projekt SHIVA geht mit den Schwerpunkten der nicht-funktionalen Chip-Infrastruktur und der formalen Analyse von Sicherheitseigenschaften über diese Projekte hinaus.

II. Projektziele

Im Projekt SHIVA werden für die Hardware von IKT-Systemen Entwurfs- und Verifikationsmethoden entwickelt, um auf Chipebene die folgenden Sicherheitseigenschaften zu garantieren:

A) Ausschluss einer beabsichtigten oder unbeabsichtigten Manipulation des Systems durch Hardware-Infrastruktur

Sowohl aus Sicherheits- als auch aus Lizenzgründen ist es erforderlich zu verhindern, dass ein Anwender die Hardware so beeinflussen kann, dass sie außerhalb des spezifizierten Verhaltens betrieben wird. Ein bekanntes Beispiel sind hier Manipulationen von Steuereinheiten zum Zweck des Fahrzeug-Tunings, welche zur Gefährdung des Betreibers und der Umwelt führen. Entsprechende Gefährdungen finden sich auch in den Bereichen Fertigungstechnik und Medizintechnik. Noch schwerwiegender sind jedoch Manipulationen von dritter Seite, um mutwillig zu schädigen. Auch hierfür finden sich inzwischen Beispielszenarien der Sabotage unterschiedlicher Anwendungen in der Presse.

B) Ausschluss der unzulässigen Beobachtung interner Daten, verwendeter Verfahren und Prozesse durch Infrastruktur

Die Daten, Verfahrensabläufe und Prozessparameter, die IKT-Systeme im industriellen Fertigungsumfeld (Industrie 4.0)

oder in medizinischen Anwendungen verarbeiten, sind vertraulich und müssen entsprechend geschützt werden. Sicherheitsplattformen sind derzeit von Halbleiterherstellern angekündigt und zum Teil verfügbar, welche sowohl die Authentifizierung als auch eine „Ende-zu-Ende“ Verschlüsselung unterstützen und dadurch einen sicheren funktionalen Zugriff auf die Prozessoren und die restliche Hardware erlauben. In diesen Ansätzen wird der Chip als Endpunkt betrachtet, aber in aller Regel wird innerhalb der Prozessoren aus Performanzgründen mit Klardaten gearbeitet und lediglich der Speicherinhalt verschlüsselt. Der physische Zugriff auf den Chip und seine integrierte Infrastruktur zur Zuverlässigkeit, Diagnose und Wartung eröffnen viele Möglichkeiten für Seitenangriffe, die ausgeschlossen werden müssen.

C) Schutz des geistigen Eigentums an der Hardware

Anwendungsspezifische Schaltungen und FPGA-basierte rekonfigurierbare Systeme enthalten geistiges Eigentum, das vor Missbrauch und Weitergabe geschützt werden muss, selbst wenn der Zugriff durch den rechtmäßigen Besitzer eines Systems erfolgt. Die Hardware sollte ein Ausforschen der Struktur und sogenanntes „reverse Engineering“ mit funktionalen Mitteln oder über unautorisierte Zugriffe mittels der Infrastruktur nicht gestatten. Zusätzlich dürfen die implementierten Strukturen auch nicht durch Dritte so geändert werden können, dass ungewollte, sicherheitsgefährdende Funktionen versteckt ausgeführt werden. Sogenannte „Trojaner“ sind in der Vergangenheit in konfigurierbare Hardware-Systeme eingeschleust worden, und sogar anwendungsspezifische fremdgefertigte Schaltungen können unter Umständen nicht vertrauenswürdig sein. Schließlich werden sichere Identifikationsverfahren benötigt, die ein Modul bzw. einen Chip eindeutig erkennen.

Eine besondere Schwierigkeit, diese drei Ziele zu erreichen, verursacht nicht-funktionale Chip-Infrastruktur, wie z.B. JTAG-Ausstattung nach IEEE Std 1149, Testinfrastrukturen nach IEEE Std 1500, rekonfigurierbare Scan-Netze (RSNs) nach IEEE Std 1687, proprietäre Strukturen für verbesserte Testbarkeit, Debug, Fehlertoleranz, Power-Management und vieles mehr. Während ohne diese Infrastruktur der sichere, wirtschaftliche und zuverlässige Betrieb eines Hardware-Systems nicht möglich ist, kann sie das System durch zahlreiche Angriffsmöglichkeiten verwundbar machen.

Das Projekt SHIVA wird von der Baden-Württemberg Stiftung im Rahmen des Forschungsprogramms IKT-Sicherheit seit Anfang 2016 über einen Zeitraum von drei Jahren gefördert. Es ist ein Kooperationsprojekt zwischen dem Institut für Technische Informatik der Universität Stuttgart (Prof. Dr. Wunderlich, Projektkoordination) und dem Lehrstuhl für Rechnerarchitektur der Universtität Freiburg (Prof. Dr. Becker).

III. Umsetzung und Methodik

Im Projekt SHIVA sollen die drei oben genannten Ziele sowohl durch Entwurfsmethoden für aktuelle und kommende rekonfigurierbare Chip-Infrastruktur als auch durch Zertifizierungstechniken für den Nachweis der geforderten Sicherheitseigenschaften, wie in Bild 1 dargestellt, erreicht werden. Dies schließt sowohl ein modulares Zugriffsrechtemanagement ein, als auch Methoden, die die Integrität und Identität von Hardware sicherstellen. Um dynamisch unterschiedliche Zugriffsrechte durchzusetzen, ist eine Authentifizierung und Autorisierung nötig. Eine hardwarenahe Umsetzung solcher Verfahren erlauben physikalisch unklonbare Funktionen (*PUF* [1]). PUFs ermitteln das Ergebnis einer Anfrage/Antwort-Autorisierung (Challenge-Response) anhand von physikalischen Eigenschaften einer Schaltungsinstanz (z.B. der Frequenz eines Ringoszillators [2], [3]) und versehen so jede Instanz mit einer ihr eigenen „Signatur“. Im industriellen Kontext werden PUFs u.A. zur Absicherung von Smartcards eingesetzt.

Die Sicherheitseigenschaften der Hardware sollen durch Einsatz formaler Methoden bewiesen werden. Diese Verfahren sollen dabei durch Ausnutzung von Entwurfsregeln, welche den Aufwand der Beweisführung reduzieren können, auch für komplexe Systeme mit hoher sequentieller Tiefe skalieren.

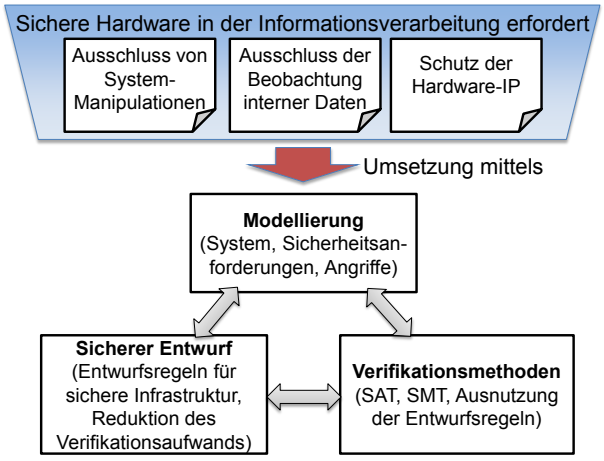

Abbildung 1. Projektziele und geplante Umsetzung

Das Projekt befasst sich maßgeblich mit der Vertrauenswürdigkeit (Security) des Hardware-Systems. Darüber hinaus ist aber auch die Betriebssicherheit (Safety) des Systems gefährdet, falls Angriffe und Systemmanipulation zu unspezifiziertem oder sogar gefährlichem Systemverhalten führen. Entsprechend ist der Nachweis von Sicherheitseigenschaften auch im Hinblick auf die Einhaltung von Standards der Betriebssicherheit (z.B. IEC 61508 und ISO 26262) von höchster Relevanz.

Erste Forschungsergebnisse im Bereich der Verifikation werden im Folgenden kurz beschrieben.

IV. Verifikation von Sicherheitseigenschaften von RSNs

In [4], [5] und [6] wurden vor kurzem Architekturen für sichere rekonfigurierbare Scan-Netze (RSNs) vorgestellt, die den Zugriff auf eine Untermenge von Instrumenten der Chip-Infrastruktur über ein Zugriffsrechtemanagement steuern. Dabei muss für die Überprüfung der Zugriffsrechte entweder ein geheimer Schlüssel im Klartext an den Chip übertragen

werden [4] oder eine Authentifizierung nach einem sichereren Anfrage/Antwort-Protokoll durchgeführt werden [6].

Um sicherzustellen, dass solche Architekturen den Zugriffsschutz korrekt umsetzen, ist eine formale Verifikation des Entwurfs und der geforderten Sicherheitseigenschaften unumgänglich. Versuche in [7] haben jedoch gezeigt, dass bestehende Verfikationswerkzeuge nur für kleine RSNs verwendet werden können. Bei taktgenauer Modellierung größerer RSNs führt die hohe sequentielle Tiefe insbesondere von Schiebeoperationen (hunderte oder tausende Takte für eine Schiebeoperation) zu Skalierbarkeitsproblemen.

Im Rahmen des Projekts SHIVA wurde ein neues domänenspezifisches Verfahren für den Nachweis von Sicherheitseigenschaften in RSNs mit hoher sequentieller Tiefe und kombinatorischen Abhängigkeiten [8] entwickelt. Im Folgenden werden kurz die Struktur von RSNs und der Stand der Forschung umrissen. In den Abschnitten IV-C und IV-D werden die vorgeschlagene Methode beschrieben und erste Ergebnisse diskutiert.

A. *Rekonfigurierbare Scan-Netze (RSNs)*

RSNs sind rekonfigurierbare Architekturen für den seriellen Zugriff auf Instrumente der Chip-Infrastruktur. RSNs nach IEEE Std 1687 [9] haben je einen primären Schiebeeingang und -ausgang. Wie im Beispiel in Bild 2 gezeigt, bestehen sie aus Schiebesegmenten bzw. -registern, Multiplexern auf den Schiebepfaden und Steuerlogik.

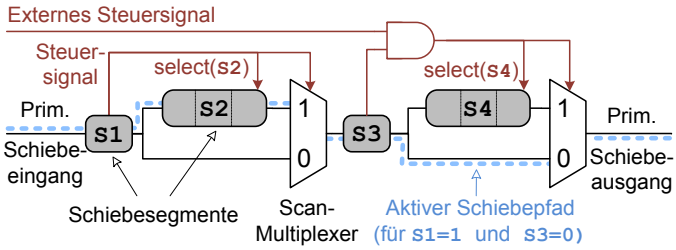

Abbildung 2. Beispiel eines rekonfigurierbaren Scan-Netzwerks

Bild 3 zeigt die Struktur eines Schiebesegments, welches ähnlich zu einem Datenregister in JTAG ein Capture- und ein Shift-Steuersignal besitzt, mit dem Daten aus einem angeschlossenem Instrument in das Segment übernommen (capture) werden können oder Daten aus dem Segment heraus- und hineingeschoben (shift) werden können. Ist das select-Signal eines Segments 0 (passiv), dann bleibt der Zustand des Segments während einer Shift- oder Capture-Operation unverändert. Optional kann ein Segment ein Schattenregister besitzen, das über ein Update-Signal gesteuert die Daten des Segments übernimmt und während des Schiebens stabil speichert. Dies ermöglicht die bidirektionale Kommunikation mit Instrumenten und die Erzeugung interner Steuersignale (intern generierte select-Signale, Multiplexer-Adress-Signale etc.), welche während des Schiebens stabile Werte haben müssen.

Multiplexer auf den Schiebepfaden erlauben, den Schiebepfad über die Steuerlogik zu rekonfigurieren, so dass z.B.

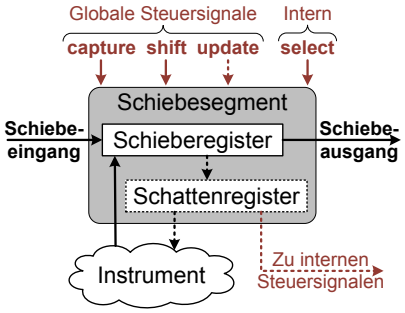

Abbildung 3. Aufbau eines Schiebesegments

durch hierarchische Schachtelung die Zugriffszeit auf Instrumente optimiert wird. Die Steuerlogik umfasst kombinatorische Funktionen der Zustände von Schiebesegmenten und externen bzw. internen Steuersignalen.

Der Lese-/Schreibzugriff auf Segmente besteht aus einer Capture-, Shift- und Update-Phase (CSU). Dabei sind nur die Segmente des RSNs aktiv, deren select-Signal 1 ist. Der aktive Schiebepfad ist ein nicht-zirkulärer Pfad vom primären Schiebeeingang über Segmente und Multiplexer zum Ausgang. Die Segmente auf dem aktiven Schiebepfad müssen aktiv sein (select ist 1) und das Adress-Signal der Multiplexer muss den Eingang mit einem aktiven Segment auswählen. Die select-Signale der anderen Segmente sind auf 0 gesetzt. So können Daten im RSN entlang des aktiven Pfads wohldefiniert gelesen und geschrieben werden.

B. *Stand der Forschung: Verifikation*

Die Korrektheit des Entwurfs einfacher statischer Schiebeketten kann durch eine topologische Traversierung der Netzliste (Korrektheit der Verbindungen) [10] durchgeführt werden. Für JTAG-basierte Architekturen wurde die Validierung der Funktion durch vierwertige Logiksimulation vorgeschlagen [11]. Für die Überprüfung des Verhaltens von IEEE 1500 Test-Wrappern eignet sich der Einsatz von constrained-random Simulationsumgebungen [12]. Mittels symbolischer Simulation kann die Äquivalenz von statischen Schiebenetzen auf unterschiedlichen Abstraktionsebenen verifiziert werden [13].

Diese Methoden können jedoch nicht auf die viel komplexeren RSNs angewendet werden. In [7] wurde eine formale Modellierung für RSNs eingeführt, die das temporale Verhalten abstrahiert und damit die Skalierbarkeit auch für umfangreiche Entwürfe gewährleistet. Die Grundidee dieser Abstraktion ist die Zusammenfassung der einzelnen Takte einer CSU-Operation in einen einzelnen atomaren Zustandsübergang des RSNs. Diese Abstraktion ist also nicht taktgenau, sondern CSU-genau (*CSU Accurate Model, CAM*). Darauf basierend wurde ein Bounded Model Checking (BMC) Verfahren entwickelt, welches Eigenschaften des Entwurfs bis zu einer gewissen temporalen Entwicklung beweisen oder widerlegen kann.

BMC eignet sich allerdings prinzipiell nicht zum Nachweis von Sicherheitseigenschaften, da hierfür *alle* erreichbaren Zustände untersucht werden müssen.

C. Verifikation des Zugriffsschutzes

Die hier vorgestelle Methode zur Verifikation des Zugriffsschutzes in RSNs verwendet die CSU-genaue temporale Abstraktion (CAM) aus [7] zur Modellierung und Craig Interpolation [14], um eine unbeschränkte Modellprüfung zu realisieren.

Die Verifikationsmethode ist in Bild 4 dargestellt. Eingabe ist die Beschreibung eines sicheren RSNs, für welches der Zugriffsschutz von Schiebesegmenten bzw. Instrumenten auf logischer Ebene verifiziert werden soll. Kommerzielle Tools erlauben es automatisch eine Beschreibung auf Register-Transfer-Ebene (RT) aus einer gegebenen Netzliste auf Gatterebene zu extrahieren [15]. Diese RT-Beschreibung wird anschließend analysiert und das CAM konstruiert. Die Transitionsrelation T, welche die möglichen Zustandstransitionen einer einzelnen CSU-Operation im RSN modelliert, wird als Boolesche Formel in konjunktiver Normalform (Menge von Klauseln) extrahiert. Zusätzlich werden die initialen Zustände (Reset-Zustände) c_0 der sequentiellen Elemente im RSN identifiziert und mit unären Klauseln kodiert. Dieses formale Modell wird um Randbedingungen möglicher Zugriffe und die nachzuweisenden Sicherheitseigenschaften (geschützte Register R) ergänzt.

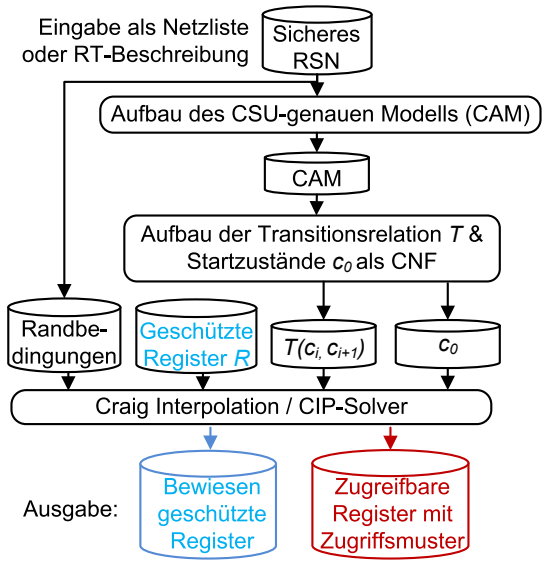

Abbildung 4. Verifikationsmethode

Die aufgestellen Klauseln sind Eingabe des CIP-Solvers, der mittels Craig Interpolation die von c_0 erreichbaren Zustände überapproximiert und auf die Sicherheitseigenschaften untersucht. Im Rahmen dieser Arbeit wird das Modell mit dem speziell auf Schaltkreise abgestimmten „CIP“-Solver [16] analysiert. Dieser Solver führt eine *unbeschränkte* Modellprüfung durch, die zum Beweis von Sicherheitseigenschaften nötig ist.

In einem sicheren RSN soll der Zugriff auf ein geschütztes Register $r \in R$ nur möglich sein, wenn die zugreifende Komponente sich autorisiert hat, typischerweise durch Vorweisen eines geheimen Schlüssels in Klartext oder kodiert in einem Anfrage/Antwort-Protokoll. Hier untersuchen wir die Möglichkeit eines Lese- oder Schreibzugriffs auf ein Segment r (Instrument) des RSNs, der ohne Vorweisen des korrekten Schlüssels ausführt werden kann und damit den Zugriffsschutz verletzt. Zu diesem Zweck werden zusätzliche Klauseln aus der RSN-Beschreibung extrahiert, um die verlangten Randbedingungen zu modellieren, zum Beispiel das Erzwingen eines falschen Schlüssels oder Geheimnisses.

Die Eigenschaft P, die widerlegt werden soll, drückt aus, dass ein Zugriff auf r in der aktuellen CSU-Operation nicht möglich ist, r also geschützt ist. Um diese Eigenschaft zu widerlegen sucht der CIP-Solver nach einer Folge von CSU-Operationen, die schließlich auf Register r zugreifen ohne den korrekten Schlüssel zu erfordern. Wird eine solche Sequenz gefunden, ist r nicht geschützt. Die berechnete Zugriffssequenz stellt einen Angriff auf die Infrastruktur dar.

Wenn der CIP-Solver bei der Berechnung der erreichbaren Zustände einen Fixpunkt erreicht und das Register r in den Zwischenschritten nicht erreichbar (Teil des aktiven Scan-Pfads) war, endet die Suche. In diesem Fall ist r in der Tat geschützt und nicht ohne den Schlüssel zugreifbar.

D. Ergebnisse

Die Verifikationsmethode wird auf zwei unterschiedliche sicheren RSN-Architekturen angewandt. Die Schaltungen wurden von den Benchmarks aus [17] abgeleitet. Die vorgestellte Methode wird genutzt, um auf Logikebene nachzuweisen, dass ein Zugriff auf geschützte Schiebesegmente nicht möglich ist, wenn der geheime Schlüssel nicht von der zugreifenden Komponente vorgewiesen werden kann. Zusätzlich wird gezeigt, wie der Zugriffsschutz der zwei Architekturen unter Annahme von potentiell beabsichtigten oder unbeabsichtigten RSN-Modifkationen untersucht werden kann. Diese RSN-Modifkationen können Entwurfsfehler oder Defekte der Hardware, Angriffe basierend auf Fehlerinjektion oder Hintertüren (Backdoors) im Entwurf darstellen.

Die erste untersuchte Architektur [4] basiert auf sog. *Locking SIBs* (LSIBs). Ein Segment Insertion Bit (SIB) ist ein rekonfigurierbarer Bypass in RSNs, der den Zugriff auf untergeordnete Schiebesegmente oder Subnetze steuert und damit hierarchische Strukturen erlaubt. In LSIB-basierten Architekturen werden durch Schließung der LSIBs der Zugriff auf die (untergeordneten) geschützten Schiebesegmente oder Subnetze unterbrochen. Ein LSIB besitzt ein spezielles Eingangssignal, das diese Unterbrechnung erzwingt und somit die geschützten Elemente unerreichbar macht [4]. Um auf ein geschütztes Element zuzugreifen muss der entsprechende LSIB geöffnet werden. Dazu muss ein vordefinierter (geheimer) Schlüssel an ausgewählte Positionen in Schiebesegmenten im RSN geschrieben werden. Eine kombinatorische Funktion prüft, ob der geforderte Wert in den verteilten Positionen vorliegt und öffnet gegebenenfalls den LSIB. Wir benutzen die vorgestellte Verifikationsmethode um zu verifizieren, dass ein Zugriff auf geschützte Elemente nicht möglich ist, solange die Werte an den ausgewählten Positionen nicht mit dem Schlüssel übereinstimmen.

Die zweite Architektur [6] verwendet sog. *Secure SIBs* (SSIBs). In SSIB-basierten Architekturen werden Schiebesegmente durch SSIBs geschützt. Die Konfiguration der SSIBs ist nur über eine separate sichere Schiebekette möglich. Der Zugriff auf die SSIBs und deren Entsperrung erfordert die vollständige und korrekte Durchführung einer Autorisierung durch ein Anfrage/Antwort-Protokolls. Da die Verschlüsselungsschaltung, welche für das Autorisierungsprotokoll genutzt wird, separat vom RSN ist, wird im CAM nur ihre Schnittstelle abgebildet. Wir verifizieren, dass ein Zugriff auf die geschützten Elemente nicht möglich ist, wenn das *responseOK* Signal, welches eine erfolgreiche Autorisierung anzeigt, deaktiviert ist. *responseOK* ist ein Ausgangssignal der Verschlüsselungsschaltung, der durch konventionelle taktgenaue Verifikationstechniken analysiert werden kann.

Die Strukur des sicheren RSNs wird auf RT-Ebene erzeugt. In jedem RSN werden zufällig fünf Schiebesegmente ausgewählt und durch mit den oben beschriebenen Verfahren geschützt. Zusätzlich zu den fehlerfreien RSNs generieren wir mutierte Beschreibungen, die Entwurfsfehler (falsche oder vertauschte Signal-Verbindungen, nicht verbundene Signale), Hardware-Defekte oder beabsichtigte Modifikationen der Architektur, wie im Folgenden beschrieben, darstellen:

- Das Reset-Signal eines SIBs, SSIBs oder LSIBs ist nicht verbunden oder sein Reset-/Start-Wert ist invertiert.
- Die kombinatorische Logik auf dem Steuersignal eines LSIB ist ergänzt um die Disjunktion mit der Konjunktion zufällig gewählten verteilten Bits in Schiebesegmenten im RSN.
- Das *responseOK*-Signal in der SSIB-Architektur ist ergänzt um die Disjunktion mit der Konjunktion von zufällig gewählten verteilten Bits in Schiebesegmenten im RSN.

Die letzten beiden Mutationen können einen Zugriff auf geschützte Segmente ermöglichen, ohne dass der korrekte Schlüssel benötigt wird. Allerdings ist eine unvollständige simulationsbasierte Validierung grundsätzlich nicht in der Lage, solche Fälle garantiert aufzudecken.

Das Mutationsexperiment wird für jeden Schaltkreis und Mutationstyp zehnmal wiederholt. Der Ort der Mutation wird zufällig in der RT-Beschreibung des RSNs gewählt. Dann wird der Zugriffsschutz zu den fünf Segmenten analysiert.

Der Verifikationsalgorithmus wurde in C++ implementiert und auf einem Kern eines Intel Xeon CPU X5680 Prozessors (3,33GHz) ausgeführt. Der vom Algorithmus benutzte Speicher überschreitet 2GB nicht.

Tabelle I fasst die Ergebnisse zusammen. Für jedes Benchmark-RSN ist die Anzahl der Schiebesegmente in der zweiten Spalte gegeben. The Gesamtzahl der Register-Bits liegt zwischen 1.416 (u226) und 97.984 Bits (p93791). Spalte 'v' zeigt den Prozentsatz der Zugriffsschutz-Verletzungen in den Experimenten, die aus den Mutationen der RSN-Entwürfe resultieren. Spalte 'd' gibt die durschnittliche Tiefe (Anzahl abgerollter Zeitfenster) an, die vom CIP-Solver benötigt wird, um die Sicherheitseigenschaft zu beweisen oder zu widerlegen. Diese Zahl entspricht nicht notwendigerweise der Anzahl an CSU Operationen, die nötig sind um die Eigenschaft zu widerlegen, da der CIP-Solver die erreichbaren Zustände in der Analyse überapproximiert. In Spalte 't' steht die durchschnittliche Laufzeit (in Sekunden) für die Verifikation der fünf geschützten Schiebesegmente. Die Laufzeit ist in allen Fällen kleiner als 1,15 Sekunden, es gibt keine Abbrüche. Spalte 6 bis 8 geben die entsprechenden Werte für SSIB-basierte RSNs an.

Tabelle I
ERGEBNISSE FÜR DIE VERIFIKATION DES ZUGRIFFSSCHUTZES

RSN	#Scan	LSIB			SSIB		
	Reg.	v [%]	d	t [s]	v [%]	d	t [s]
u226	40	14,3	3,0	0,055	33,3	3,9	0,088
d281	50	16,2	3,0	0,060	28,6	3,9	0,094
d695	157	17,1	3,1	0,177	47,6	4,0	0,299
h953	46	16,2	3,0	0,060	28,6	3,9	0,085
g1023	65	16,2	3,0	0,065	16,2	4,2	0,110
f2126	36	13,3	3,0	0,048	23,8	3,8	0,073
q12710	21	12,4	3,0	0,013	10,5	4,2	0,052
p22810	254	18,1	3,1	0,297	42,9	4,1	0,467
p34392	103	17,1	3,2	0,125	4,8	4,7	0,150
p93791	588	19,0	3,1	0,715	47,6	4,1	1,149
t512505	128	19,0	3,1	0,153	33,3	4,0	0,233
a586710	32	12,4	3,0	0,043	23,8	4,1	0,073

Die maximale Anzahl an Klauseln der Transitionsrelation T ist 33.547 für das SSIB-basierten RSN p93791 mit der Mutation des *responseOK* Signals. Die maximale vom CIP-Solver benötigte Tiefe ist 6 für die LSIB-Architekturen und 8 für die SSIB-Architekturen.

Die Ergebnisse dieser Experimente zeigen, dass diese Verifikationsmethode auch auf große RSNs anwendbar ist, selbst wenn diese aufgrund des Zugriffsschutzes zusätzliche komplexen Abhängigkeiten aufweisen. Die sehr geringe Laufzeit und das stabile Verhalten der vorgeschlagenen unbeschränkten Modellprüfung erlauben weiterhin, den Zugriffsschutz von RSN-Architekturen für eine große Anzahl unterschiedlicher Szenarien zu untersuchen.

V. ZUSAMMENFASSUNG

Das vorgestellte Projekt SHIVA erforscht Entwurfs- und Verifikationsmethoden zur Steigerung der Sicherheit zukünftiger Hardware-Systeme, welche die Grundlage für IKT-Sicherheit darstellen.

Eine skalierbare Verifikationsmethode des Zugriffsschutzes in sicherer rekonfigurierer Chip-Infrastruktur wurde kurz vorgestellt. Diese Methode erlaubt den effizienten Nachweis von Sicherheitseigenschaften für Infrastrukturen mit Schiebenetzen mit sehr hoher sequentieller Tiefe. Weiterhin erlaubt die Modellierung die Beschreibung und Untersuchung von Fehlern im Entwurfsprozess, wie z.B. eine fehlerhafte Implementierung des Zugriffsschutzes, als auch von Angriffen, z.B. basierend auf Fehlerinjektionen.

DANKSAGUNG

Diese Arbeit ist Teil des Projekts SHIVA und wurde von der Baden-Württemberg Stiftung im Rahmen des Forschungsprogramms „IKT-Sicherheit“ unterstützt.

LITERATUR

[1] G. E. Suh und S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. ACM/IEEE Design Automation Conference (DAC)*, 2007, pp. 9–14.

[2] A. Maiti und P. Schaumont, "Improved ring oscillator PUF: an FPGA-friendly secure primitive," *Journal of Cryptology*, vol. 24, no. 2, pp. 375–397, 2011.

[3] L. Feiten, A. Spilla *et al.*, "Implementation and analysis of ring oscillator PUFs on 60 nm Altera Cyclone FPGAs," *Information Security Journal: A Global Perspective*, vol. 22, no. 5-6, pp. 265–273, 2013.

[4] J. Dworak, A. Crouch *et al.*, "Don't Forget to Lock your SIB: Hiding Instruments using P1687," in *Proc. IEEE Int'l Test Conf.*, 2013, paper 6.2.

[5] R. Baranowski, M. A. Kochte und H.-J. Wunderlich, "Access Port Protection for Reconfigurable Scan Networks," *Journal of Electronic Testing (JETTA)*, vol. 30, pp. 711–723, 2014.

[6] R. Baranowski, M. A. Kochte und H.-J. Wunderlich, "Fine-grained access management in reconfigurable scan networks," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 34, no. 6, pp. 937–946, June 2015.

[7] R. Baranowski, M. A. Kochte und H.-J. Wunderlich, "Reconfigurable Scan Networks: Modeling, Verification, and Optimal Pattern Generation," *ACM Trans. Design Automation of Electronic Systems (TODAES)*, vol. 20, no. 2, pp. 30:1–30:27, 2015.

[8] M. A. Kochte, R. Baranowski *et al.*, "Formal Verification of Secure Reconfigurable Scan Network Infrastructure," in *Proc. IEEE European Test Symposium (ETS)*, May 2016.

[9] "IEEE Std. 1687-2014 – IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device," Dec. 2014, IEEE Computer Society.

[10] R. Fisher, "Method and Apparatus to Check the Integrity of Scan Chain Connectivity by Traversing the Test Logic of the Device," Nov. 2002, US Patent App. 10/300,513.

[11] K. Melocco, H. Arora *et al.*, "A Comprehensive Approach to Assessing and Analyzing 1149.1 Test Logic," in *Proc. IEEE Int'l Test Conference (ITC)*, 2003, pp. 358–367.

[12] I. Diamantidis, T. Oikonomou und S. Diamantidis, "Towards an IEEE P1500 Verification Infrastructure: A Comprehensive Approach," in *Proc. IEEE Int'l Workshop on Infrastructure IP*, 2005.

[13] H. B. Kamepalli, P. Sanjeevarao und C.-J. Park, "Scan Chain Verification Using Symbolic Simulation," Patent, 2006, US Patent App. 7,055,118.

[14] W. Craig, "Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory," *J. Symbolic Logic*, vol. 22, no. 3, pp. 269–285, 1957.

[15] Mentor Graphics Corporation, "Automation of the IEEE 1687 Standard: Tessent IJTAG," 2014, datasheet. [Online]. Available: www.mentor.com

[16] S. Kupferschmid, M. Lewis *et al.*, "Incremental Preprocessing Methods for Use in BMC," *Formal Methods in System Design*, pp. 1–20, 2011.

[17] R. Baranowski, M. A. Kochte und H.-J. Wunderlich, "Modeling, Verification and Pattern Generation for Reconfigurable Scan Networks," in *Proc. IEEE Int'l Test Conf. (ITC)*, 2012, paper 8.2.