Physically Unclonable Functions (PUFs) als neue Technologie in Sicherheitsanwendungen

Andrea Kolberger [a], Ingrid Schaumüller-Bichl [a], Martin Deutschmann [b]

[a] FH OÖ Fakultät für Informatik, Kommunikation und Medien, Hagenberg, Softwarepark 11, A-4232 Hagenberg, AUSTRIA

[b] TECHNIKON Forschungs- und Planungsgesellschaft mbH, Burgplatz 3a, A-9500 Villach, AUSTRIA

KURZFASSUNG/ABSTRACT:

Physically Unclonable Functions (PUFs) sind eine neuartige, vielversprechende Technologie in kryptografischen Anwendungsgebieten. Die Idee dahinter ist die Nutzung von inhärenten physikalischen Eigenschaften von integrierten Schaltkreisen (IC), die mit dem menschlichen Fingerabdruck vergleichbar sind. Im Zuge eines FIT-IT Forschungsprojektes wurden neuartige kryptografische low-cost Algorithmen und Protokolle entwickelt, um die Stabilität und Zuverlässigkeit von PUF-Messungen zu erhöhen. Außerdem wurde ein Protection Profile zur Sicherheitsevaluierung von PUF-basierten IT-Produkten erstellt.

1 EINLEITUNG

Der Schutz von geistigem Eigentum (Intellectual Property – IP) und die Absicherung gegen Reverse Engineering gehören zu den größten Herausforderungen am IT-Sektor. Die Angriffe auf kryptografische Systeme werden stets ausgefeilter und die Schutzmaßnahmen müssen immer umfangreicher werden und neue Wege einschlagen. Die Absicherung von Sicherheitsmodulen erfolgt meist durch den Einsatz von kryptografischen Verfahren wie symmetrischen oder asymmetrischen Verschlüsselungsalgorithmen oder digitalen Signaturen. Die Sicherheit solcher Systeme beruht auf der Geheimhaltung eines kryptografischen Schlüssels und dieser ist ausgeklügelten Hardware-Angriffen, wie Seitenkanal- oder Timingattacken, ausgesetzt.

Einen neuen Weg Sicherheit von sensiblen Daten zu gewährleisten bietet der Einsatz von Physically Unclonable Functions (PUFs). Bei dieser völlig neuen Technologie ist ein Speichern des kryptografischen Schlüssels im System nicht mehr nötig. Es werden inhärente physikalische Parameter der bestehenden Hardware bereits im Fertigungsprozess ausgelesen und in Form eines digitalen Fingerabdruckes des ICs (Integrated Circuits) verwendet – das Gerät selbst ist der Schlüssel. Dies erschwert Angriffe auf die Hardware um ein Vielfaches. Die neuartigen Sicherheitsfeatures von PUFs ermöglichen somit das Sicherheitsniveau von kryptografischen Anwendungen anzuheben und bieten gleichzeitig kostengünstige Implementierungsmöglichkeiten.

Im Rahmen des Forschungsprojektes CODES wurden neuartige kryptografische low-cost Algorithmen und Protokolle für rekonfigurierbare PUFs entwickelt. Diese wurden auch in einem Prototyp implementiert. Zusätzlich wurde die Basis für Sicherheitsevaluierungen und Pre-Zertifizierungen in realer Umgebung nach Common Criteria geschaffen. Kapitel 2 geht zunächst auf die neuartige Technologie von PUFs ein und erklärt deren Eigenschaften, Qualitätsmerkmale und mögliche Einsatzgebiete, während Kapitel 3 die Ziele und Ergebnisse des CODES Projektes erläutert und die darin gewählte Vorgehensweise beschreibt. Das Paper schließt mit einer Zusammenfassung in Kapitel 4 mit Ausblick auf mögliche Future Work.

2 PHYSICALLY UNCLONABLE FUNCTIONS

Physically Unclonable Functions (PUFs) [14] nutzen die inhärenten physikalischen Eigenschaften von Elektronikbauteilen, die sich während der Herstellung ergeben und für jeden Bauteil einzigartig sind – also eine Art „digitaler Fingerprint“. Diese Eigenschaften können in der Produktion nicht beeinflusst bzw. kontrolliert werden und sind damit auf physikalischer Ebene nicht zu klonen.

Die grundlegende Funktionsweise besteht in einem Challenge-Response-Verfahren. Ein PUF erhält eine Challenge und generiert aufgrund seiner inhärenten physikalischen Eigenheiten eine Response. Nur diese eine PUF-Instanz kann eine bestimmte Response zu einer bestimmten Challenge liefern, da die physikalischen Eigenschaften an diese eine Instanz gebunden sind. Da es sich um physikalische Messungen handelt, sind PUF-generierte Responses allerdings fehlerbehaftet und werden als „noisy" bezeichnet. Je nach Anwendungsgebiet kann es ausreichend sein, wenn eine PUF-generierte Response – bis auf einige wenige Fehler – einer Referenz-Response sehr ähnlich ist (z.B. bei Authentifizierungsverfahren). Dies trifft aber nicht für kryptografische Anwendungen zu, bei denen es zwingend erforderlich ist, dass die Bitfolge einer Response vollständig ident ist, wie etwa bei der Generierung eines kryptografischen Schlüssels. Um PUFs für die Schlüsselerzeugung einzusetzen, werden fehlerkorrigierende Codes angewendet. Zudem werden häufig Hash-Funktionen eingesetzt um eine uniforme Verteilung des Schlüsselbitmusters zu erreichen.

PUF-basierte Schlüsselerzeugung umgeht eine der größten Herausforderungen von kryptografischen Systemen – die Geheimhaltung des privaten Schlüssels. In der Praxis ist es häufig üblich Schlüsselmaterial direkt am Chip (oft im sogenannten non-volatile memory) abzulegen. Ein Angreifer könnte damit, wenn Zugriff zum Gerät besteht, (offline) Versuche starten Schlüsselinformationen auszulesen. Solche Angriffe sind oftmals erfolgreich und daher ist die Krypto-Community bestrebt Gegenmaßnahmen zu finden. PUF-basierte Schlüsselerzeugung funktioniert in zwei Schritten. Zunächst wird ein PUF in einer sicheren Umgebung „enrolt", wobei der PUF erstmalig ausgelesen und sogenannte (nicht sensitive) Helper Daten (helper data) generiert werden, die ohne weiteres am Chip abgespeichert werden können. Während des Betriebes (in einer unsicheren Umgebung) wird der PUF bei Bedarf ausgelesen und nur das Zusammenspiel aus korrektem PUF und zugehörigen Helper Daten erlaubt eine Rückgewinnung des Schlüssels. Ein entscheidender Baustein sind hier fehlerkorrigierende Codes, da PUF-Messungen fehlerbehaftet sein können. Ein Angreifer hat also im offline Modus kein Angriffsziel, da die einzigen statischen Daten die Helper Daten sind, die keine sensiblen Inhalte preisgeben.

2.1 Eigenschaften

Physically Unclonable Functions können mit folgenden Eigenschaften beschrieben werden [14]:

- PUFs sind (physikalische) Ein-Weg-Funktionen. Für einen gegebenen PUF und eine Challenge x ist es einfach $PUF(x)$ zu berechnen. Für einen gegebenen PUF und eine Response y ist es allerdings schwierig, ein x zu finden, sodass $PUF(x) = y$.
- Eine PUF Response ist, bis auf wenige Fehler, reproduzierbar, d.h. $y = PUF(x)$.
- PUFs sind physikalisch nicht zu klonen, d.h. für einen gegebenen PUF ist es schwierig einen PUF' mit gleichen Eigenschaften, bis auf wenige Fehler, zu konstruieren, sodass $\forall x \in X : PUF'(x) \approx PUF(x)$.
- PUF Responses sind nicht vorhersehbar. Trotz Kenntnis mehrerer gültiger Challenge-Response-Pairs $Q = \{(x_i, y_i) \mid y_i = PUF(x_i)\}$ für einen gegebenen PUF ist es schwierig für eine beliebige Challenge $(x_c,.) \notin Q$ eine gültige Response, bis auf wenige Fehler, vorherzusagen, sodass $y_c \approx PUF(x_c)$.

2.2 Qualitätsmerkmale

Qualität bzw. Stabilität und Verlässlichkeit von PUFs können durch unterschiedliche Parameter charakterisiert werden. Ein entscheidender Aspekt ist die Einzigartigkeit von PUFs, d.h. unterschiedliche PUFs sollten deutlich voneinander differenzierbar sein. Dies wird charakterisiert durch die sogenannte inter-Distanz. Sehr häufig werden Responses als binäre Bitfolgen dargestellt und als Metrik wird die Hamming-Distanz herangezogen. Dann wird die inter-Distanz als Hamming-Distanz zwischen PUF Responses berechnet. Im Optimalfall erreicht man eine (relative) Hamming-Distanz von 50%, d.h. größtmögliche Unterscheidung. Analog dazu möchte man, dass das mehrfache Auslesen desselben PUFs stets die möglichst gleiche Response liefert. Die PUF-Stabilität wird mit der sogenannten intra-Distanz angegeben, welche im Opti-

malfall gegen Null gehen sollte. Durch die verrauschten Messungen ist jedoch mit einer gewissen Fehlerrate zu rechnen, die mit Hilfe von fehlerkorrigierenden Codes in den Griff zu bekommen ist. Es zeigt sich zudem, dass eine gewisse Anzahl von Bits einen Großteil der Fehler verursacht, d.h. man kann durch Messungen sogenannte „dark-Bits“ identifizieren. Häufig wird versucht diese Bits gesondert zu behandeln (z.B.: weglassen), wodurch zwar die verwendbare Response Bitfolge reduziert wird, jedoch das Fehlerverhalten deutlich verbessert werden kann. PUFs sind elektronische Bauteile und daher auch temperaturabhängig. Ein Einsatz bei Raumtemperatur zeigt meist besseres Verhalten als bei extremen Temperaturen. Dies ist beim Dimensionieren von fehlerkorrigierenden Methoden unbedingt zu berücksichtigen; im Regelfall wird für den „Worst Case“ dimensioniert. Eine zusätzliche Komponente, die viele elektronische Bauteile aufweisen, ist die Alterung. Ein PUF verändert seine Response über die Zeit marginal, dennoch kann dies zu Überschreitung von Fehlergrenzwerten führen, was wiederum bei der Dimensionierung von Fehlerkorrekturmethoden zu berücksichtigen ist.

2.3 Anwendungsgebiete

PUF-Technologie kann in zahlreichen Anwendungsgebieten genutzt werden. Im Projekt CODES wurden Einsatzzwecke in Authentifizierungsverfahren und kryptografischen Algorithmen und Protokollen berücksichtigt. Ein sehr einfacher Use Case ist die *One-Way Authentication*, bei dem sich nur der PUF-basierte Token authentifiziert. In diesem Fall kann es ausreichend sein, wenn eine generierte PUF Response einer Referenz-Response sehr ähnlich ist, d.h. auf eine PUF Response werden keine Fehlerkorrekturmechanismen angewendet. Im Unterschied dazu authentifizieren sich bei der *Mutual Authentication* [10] beide Kommunikationspartner gegenseitig. Um eine eindeutige PUF Response zu erhalten, werden fehlerkorrigierende Codes (Error Correction Codes, ECC) eingesetzt. PUF Responses können zudem herangezogen werden um symmetrische Session Keys für die weitere Kommunikation zu verschlüsseln und auf diesem Wege zwischen den Kommunikationspartnern auszutauschen. PUFs können auch eingesetzt werden um Software an eine bestimmte Hardware zu binden (*HW/SW Binding*). Ein Schlüssel, abgeleitet von einer PUF Response, wird zur Verschlüsselung der Software herangezogen, die im Speicher des Chips hinterlegt wird. Nur dieser eine Chip mit dem entsprechenden PUF ermöglicht die erfolgreiche Entschlüsselung der Software. Durch das Hinzufügen von zusätzlichen Informationen (state information) bei der Generierung von PUF-basierten Schlüsseln kann im Bedarfsfall verwendetes Schlüsselmaterial geändert werden (reconfigurable PUFs, *Key Zeroization*) [7, 11]. Diese Funktionalität könnte beispielweise bei der Übertragung von verschlüsselten Signalen (Rundfunk, Fernsehen) angewendet werden.

3 DAS PROJEKT CODES

Das Forschungsprojekt CODES (Algorithmic extraction and error correction codes for lightweight security anchors with reconfigurable PUFs) wurde von der Österreichischen Forschungsförderungsgesellschaft (FFG) im Rahmen des FIT-IT Programms gefördert. Unter der Leitung und Koordination der Kärntner Firma TECHNIKON waren die Alpen-Adria Universität Klagenfurt (Mathematisches Institut) sowie die FH OÖ Forschungs & Entwicklungs GmbH, Campus Hagenberg (Department für Sichere Informationssysteme), am Projekt beteiligt.

3.1 Projektziele

Ziel des Projektes war die Entwicklung von kryptografischen low-cost Algorithmen und Protokollen um die Zuverlässigkeit und Stabilität von PUF-Messungen zu erhöhen. Dabei soll mit Fehlerkorrekturcodes und Anti-Ageing-Mechanismen die Sicherheit von PUF-basierten IT-Produkten verbessert werden. Die entwickelten Protokolle und Verfahren sollen in einem kombinierten FPGA-ASIC Prototyp integriert werden, wofür 65nm CMOS ASICs vorhanden sind, die sechs unterschiedliche PUF Realisierungen beinhalten. Ein weiterer wesentlicher Bestandteil des Projektes war die Entwicklung eines Schutzprofils für PUF-basierte IT-Produkte um damit die Basis für Sicherheitsevaluierungen und Pre-Zertifizierungen nach den international anerkannten Common Criteria (CC) [2] zu legen.

3.2 Vorgehensweise

Unter Berücksichtigung der einzigartigen Sicherheitsfeatures und Besonderheiten von PUFs wurde zu Beginn eine umfassende Risikoanalyse für bestehende und neue PUF-basierte Verfahren durchgeführt. Dabei wurde nicht nur die Technologie an sich, sondern auch das administrative und operative Umfeld für den gesamten Lebenszyklus von PUF-basierten Produkten berücksichtigt. Im Speziellen wurden dafür verschiedene Anwendungsfälle definiert um mögliche Bedrohungen bzw. potentielle Schwachstellen zu identifizieren. Die Use Cases beziehen sich auf Authentifizierungsprotokolle sowie Verfahren zur Erzeugung von kryptografischem Schlüsselmaterial: One-Way Authentication, Mutual Authentication, Secret Key Generation and Session Key Exchange, Hardware/Software Binding und Key Zeroization.

Wie bereits in Kapitel 2.2 angeführt, gibt es unterschiedliche Qualitätsmerkmale, die PUFs charakterisieren. Diese Parameter wurden im CODES Projekt für bestimmte PUF-Typen ausgearbeitet und damit eine Dimensionierung der sogenannten Helper Data Algorithms (HDA) ermöglicht. Aus einem früheren Projekt liegen eine Reihe von konkreter PUF-Ausführungen vor. Diese Auswahl umfasst unter anderem Arbiter, Ring Oszillator und SRAM PUFs. Durch statistische Untersuchungen konnten Parameter, wie die Inter- oder die Intra-Distanz für diese PUF Typen berechnet werden. Außerdem lagen Ergebnisse aus Messungen bei anderen Temperaturbereichen (-45°C und 120°C) vor. Auf Basis dieser Daten konnten im Rahmen des CODES-Projektes die „Worst Case“-Abschätzungen der zu erwartenden Fehler berechnet und die für diesen Zweck geeigneten fehlerkorrigierenden Codes ermittelt werden.

Weiters wurde ein Schutzprofil nach Common Criteria erarbeitet, bei dessen Erstellung die funktionalen Sicherheitsanforderungen an PUF-basierte IT-Produkte zum einen den Common Criteria Part 2 [3] entnommen und zum anderen aufgrund der Erkenntnisse aus der Risikoanalyse definiert wurden. Die entwickelten Algorithmen und Protokolle sowie funktionale Sicherheitsanforderungen lieferten schließlich die grundlegenden Funktionalitäten, die in einer Prototypimplementierung realisiert werden sollten.

3.3 Ergebnisse

Die Ergebnisse der Risikoanalyse zeigten, dass das höchste Risiko bei der Implementierung der fehlerkorrigierenden Codes liegt: einerseits soll die Zuverlässigkeit der Funktionalität sichergestellt werden und andererseits dürfen die Algorithmen bzw. benötigte Zusatzinformationen (helper data) zur Rekonstruktion von Geheimnissen keine Rückschlüsse auf die PUF Response ermöglichen. Durch Ageing-Effekte oder das instabile Verhalten eines PUFs ist es notwendig starke Fehlerkorrekturmechanismen bereitzustellen, die im Stande sind PUF Responses bzw. kryptografische Schlüssel verlässlich wiederherzustellen. Weitere Risiken, die aus der Einsatzumgebung entstehen, können häufig nicht direkt von PUFs behandelt werden und sind daher durch angemessene Sicherheitsanforderungen an die Umgebung zu berücksichtigen. Ein Beispiel dafür ist, dass jeder PUF-Token mit unterschiedlichen, zufälligen und nicht vorhersehbaren Challenges „enrolt“ werden muss, um es einem Angreifer zu erschweren, gültige Challenges zu erraten. Zudem ist dafür zu sorgen, dass die Übertragung einer Datenbank mit gültigen Challenge-Response-Pairs (CRPs) hinsichtlich Vertraulichkeit und Integrität gesichert erfolgt. [13]

Die statistische Analyse von bereits vorhandenen und generierten PUF-Messungen zeigte, dass Arbiter PUFs (delay-based PUFs) [8] und SRAM PUFs (memory-based PUFs) [9] das stabilste Verhalten aufwiesen. Bei Raumtemperatur ergaben unsere Messungen folgende Durchschnittswerte:

Tabelle 1. Inter- und Intra-Distanz von Arbiter- und SRAM-PUF.

PUF Typ	Inter-Distanz	Intra-Distanz
Arbiter PUF	46.51 %	3.05 %
SRAM-PUF	49.34 %	5.21 %

Zudem konnte im Projekt untermauert werden, dass eine geringe Anzahl an Response Bits häufig für den Großteil der Fehler verantwortlich ist. Dies kann durch folgenden Plot dargestellt werden:

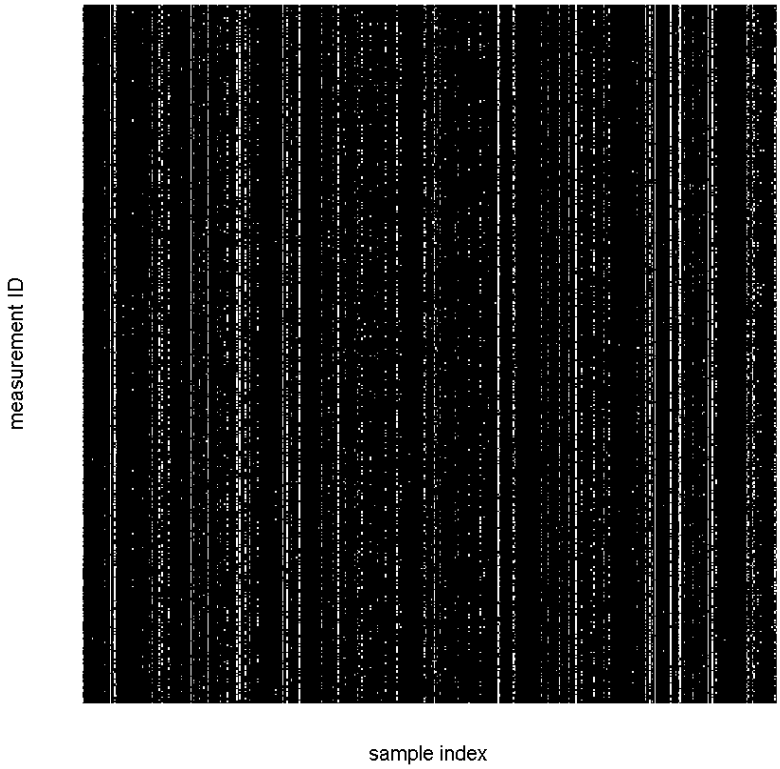

Abbildung 1. Dark Bits in SRAM PUF Readout

Wie in Abbildung 1 erkennbar, flippen sehr häufig die gleichen Bits (hellen Punkte). Die Auswertung bezieht sich auf 500 Messungen einer PUF Response (measurement ID) bei einer Response-Länge von 512 Bits (sample index). Wenn man diese „fehlerhaften“ Bits identifiziert und gesondert betrachtet (z.B. weglässt) so ist eine deutliche Reduktion der Bit Fehler (Intra-Distanz) möglich, wodurch eine schlankere Fehlerkorrektur eingesetzt werden kann.

Mit diesen Erkenntnissen konnten wir die passenden fehlerkorrigierenden Codes auswählen. Eine übliche Praxis ist die Verwendung von konkatenierten Codes. Hierbei gibt es einen (simplen) inneren Code und einen (etwas komplexeren) äußeren Code. Bei der Decodierung erreicht man dadurch den Effekt, dass ein simpler Code (z.B. Repetition Code) die Fehlerrate soweit drückt, dass der danach folgende äußere Code die verbleibenden Fehler sicher korrigieren kann. Bosch et al. belegen in ihrem Paper [1], dass diese Form der Zusammenschaltung von Codes zum einen bessere Fehlerkorrektureigenschaften ausweist und zum anderen die Komplexität des Decodierungsalgorithmus beschränkt bleibt. Im Zuge des CODES Projektes wurden unterschiedliche Kombinationen von Repetition, Reed-Miller, Reed-Solomon und BCH Codes entwickelt und implementiert, welche auf die jeweiligen Use Cases maßgeschneidert dimensio-

niert sind. Zudem konnten neuartige Ansätze entwickelt werden, bei denen Alterungseffekte in der Dekodierung implizit behandelt werden. Der grundlegende Ansatz war stets möglichst simple Codes zu verwenden um die Komplexität der Algorithmen gering zu halten.

Als Basis für Sicherheitsevaluierungen und Produktzertifizierungen wurde des Weiteren eine erste Version eines Protection Profiles für PUF-basierte Produkte [12] erstellt. Dieses enthält alle wesentlichen Inhalte, die von den Common Criteria gefordert sind und fasst sämtliche Sicherheitsanforderungen zusammen, die in einem PUF-basierten IT-Produkt zu berücksichtigen sind. Dabei wurden die funktionalen Sicherheitsanforderungen der CC um PUF-spezifische Anforderungen (Extended Components) erweitert, um damit den Besonderheiten dieser Technologie Rechnung zu tragen. Neben funktionalen Sicherheitsanforderungen wurden auch Anforderungen an die Vertrauenswürdigkeit von PUF-basierten Produkten festgelegt. Als Grundlage für die Definition der Sicherheitsanforderungen dienten dabei die Anforderungen der Common Criteria Part 2 [3] und Part 3 [4].

Basierend auf den Erkenntnissen aus der statistischen Analyse von PUF-Messungen wurden Verfahren zur Fehlerkorrektur in einer Prototypimplementierung realisiert, die zwei unterschiedliche Anwendungsfälle repräsentiert: „Gegenseitige Authentifizierung“ [10] und „Schlüsselgenerierung“ [6]. In zwei separaten Graphical User Interfaces (GUIs) können diese Anwendungen transparent für den User ausgeführt werden, wobei der Anwender die Möglichkeit zur Interaktion mit der Prototypimplementierung hat. Beispielsweise werden schrittweise die generierten und übertragenen Daten am Bildschirm ausgegeben und der Anwender kann versuchen bestimmte Daten zu manipulieren um herauszufinden, ob ein Fehlverhalten erkannt wird. Gleichzeitig wurden Mechanismen integriert, die definierte Sicherheitsanforderungen des Schutzprofils abdecken, wie bspw. das Erkennen von Replay-Attacken. Der funktionsbereite Prototyp [5] setzt sich aus drei Hardwarekomponenten zusammen:

(1) Der PC stellt beide GUIs bereit und ist mit dem FPGA Board verbunden. Fehlerkorrekturmechanismen, die auf fehlerbehaftete PUF Responses anzuwenden sind, wurden in C/C++ implementiert.

(2) Das FPGA Board stellt die Schnittstelle zwischen PC und UNIQUE ASIC Board (s.u.) zur Verfügung, sendet Challenges an das ASIC Board, kann ebenfalls notwendige Fehlerkorrekturen (in VHDL implementiert) durchführen und leitet generierte Daten an den PC weiter.

(3) Das ASIC Board ist ein Resultat aus dem früheren europäischen Forschungsprojekt UNIQUE, das im Projekt CODES genutzt werden konnte. Dieses integriert fünf Chips, von denen jeder sechs verschiedene PUF-Instanzen bereitstellt. Das ASIC Board nimmt eine Challenge vom FPGA Board entgegen, generiert die entsprechende Response und sendet diese an das FPGA Board zurück.

Fehlerkorrigierende Codes können unter Umständen sehr rechenintensiv sein. Da auf bestimmten Geräten die Rechenkapazität beschränkt sein kann (z.B. Smartcards), besteht die Möglichkeit, dass aufwändige Berechnungen auf leistungsstärkeren Geräten ausgeführt werden. Dies ist der Grund, weshalb die CODES Prototypimplementierung diese Codes sowohl PC-seitig als auch auf dem FPGA Board realisiert.

4 ZUSAMMENFASSUNG UND AUSBLICK

Die Prototypimplementierung demonstriert die erfolgreiche gegenseitige Authentifizierung eines PUF-basierten Produktes sowie die Erzeugung und Rekonstruktion von kryptografischem Schlüsselmaterial. Des Weiteren steht eine erste stabile Version eines Protection Profiles nach CC für PUF-basierte Produkte zur Verfügung und damit eine fundierte Basis für spätere Produktzertifizierungen. Erste Projektergebnisse wurden bereits bei internationalen Sicherheitskonferenzen präsentiert und eine weitere Journalpublikation ist geplant.

In Zukunft sollen weitere Anwendungsmöglichkeiten von PUF-Technologien in spezifischen Gebieten analysiert werden, die aus Sicherheitssicht von den Eigenschaften dieser Technologie besonders profitieren können.

LITERATURVERWEISE

[1] Bosch, Ch., Guajardo, J., Sadeghi, A.-R., Shokrollahi, J. and Tuyls, P. (2008): Efficient helper data key extractor on fpgas, in Elisabeth Oswald and Pankaj Rohatgi, editors, Cryptographic Hardware and Embedded Systems – CHES 2008, volume 5154 of Lecture Notes in Computer Science, pages 181–197, Springer Berlin Heidelberg.

[2] Common Criteria for Information Technology Security Evaluation, Part 1 (2012): Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4.

[3] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012

[4] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; CCMB-2012-09-003, Version 3.1, Revision 4, September 2012

[5] Deutschmann, M., Höberl, M., Petschnigg, Ch., Schaumüller-Bichl, I., Kolberger, A., Mazzoli, M. and Heuberger, C. (2014): D3.1: Hybrid FPGA ASIC prototype, Project CODES.

[6] Dodis, Y., Ostrovsky, R., Reyzin, L.and Smith, A. (2008): Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data, SIAM Journal on Computing, 38(1):97-139.

[7] Eichhorn, I., Koeberl, P., van der Leest, V.: Logically reconfigurable PUFs: memorybased secure key storage. In Proceedings of the sixth ACM workshop on Scalable trusted computing, STC '11, 59-64, New York, USA (2011)

[8] Fruhashi, K., Shiozaki, M., Fukushima, A., Murayama, T. and Fujino, T. (2011): The arbiter-PUF with high uniqueness utilizing novel arbiter circuit with Delay-Time Measurement. IEEE International Symposium on Circuits and Systems (ISCAS) 2011, pages 2325-2328.

[9] Handschuh, H. (2012): Hardware-Anchored Security Based on SRAM PUFs, Part 1. Security Privacy, IEEE, 10(3):80-83.

[10] Herrewege, A., Katzenbeisser, S., Maes, R., Peeters, R., Sadeghi, A.-R., Verbauwhede, I. and Wachsmann, Ch. (2012): Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs, In AngelosD. Keromytis, editor, Financial Cryptography and Data Security, volume 7397 of Lecture Notes in Computer Science, pages 374-389, Springer Berlin Heidelberg.

[11] Katzenbeisser, S., Kocabas, U., van der Leest, V., Sadeghi, A.-R., Schrijen, G.-J., Schröder, H., Wachsmann, Ch.: Recycable PUFs: Logically Reconfigurable PUFs. (2007)

[12] Kolberger, A., Schaumüller-Bichl, I., Brunner, V. and Deutschmann, M. (2014): Protection Profile for PUF-Based Devices, ICT Systems Security and Privacy Protection, SEC 2014, IFIP Advances in Information and Communication Technology, Vol. 428, pages 91-98, Springer Berlin Heidelberg.

[13] Kolberger, A., Schaumüller-Bichl, I. and Deutschmann, M. (2014): Risk Analysis of Physically Unclonable Functions, Communications and Multimedia Security (CMS 2014), Lecture Notes in Computer Science, Vol. 8735, pages 136-139, Springer Berlin Heidelberg.

[14] Verbauwhede, I. and Maes, R. (2010): Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions, in Ahmad-Reza Sadeghi and David Naccache, editors, Towards Hardware-Intrinsic Security, Information Security and Cryptography, pages 3-37, Springer Berlin Heidelberg.