KRYPTOTAG

Fraunhofer
AISEC

18th Crypto-Day

Fraunhofer Research Institution AISEC
July 4 and 5, 2013
Garching (near Munich), Germany

Hardware Efficient Authentication based on Random Selection

Frederik Armknecht, Matthias Hamann, Matthias Krause

University of Mannheim
Mannheim, Germany

Devices of extremely small computational power like radio frequency identification (RFID) tags are used in practice to a rapidly growing extent, a trend commonly referred to as ubiquitous computing. One of the major use-cases for such pervasive devices are authentication solutions, e.g., access control for buildings or cars, electronic passports or even human-implantable chips providing sensitive medical information about a person.

Consequently, the search for lightweight authentication protocols became an important topic in cryptography during the last years with high relevance for academia and industry.

Today, one can distinguish three main approaches for constructing lightweight authentication protocols:

1. protocols which use lightweight block ciphers like PRESENT, KATAN and KTANTAN as basic cryptographic operations,
2. protocols which employ the well-researched principle of adding biased noise to a secret linear function (i.e., the LPN problem),
3. protocols which are based on the principle of random selection, being the most recent of all three paradigms.

While almost all LPN-type protocols (approach 2) have eventually been shown to be vulnerable w.r.t. active or passive attacks, the (n, k, L)-protocol introduced in [KS09] and based on approach 3 remains yet unbroken (see [KH11] for an in-depth security analysis) . However a comparatively huge key length and the use of involved operations made a hardware-efficient implementation a challenging task so far.

In this work we introduce the $(n, k, L)^{80}$-protocol, a variant of linear authentication protocols which overcomes these problems, and analyze its security against all currently known, relevant passive and active attacks. Moreover, we present an implementation of our protocol for Field Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs) using Verilog and discuss its efficiency w.r.t. generally accepted costs metrics. The respective numbers show that the $(n, k, L)^{80}$-protocol is a viable alternative to existing solutions and is, for example, well suited for the implementation on passive RFID tags. In particular, to our knowledge, this is the first lightweight authentication protocol which can be realized at costs below 1,000 Gate Equivalents without succumbing to known active or passive attacks.

References

[KS09] M. Krause and D. Stegemann. More on the Security of Linear RFID Authentication Protocols. *Proceedings of SAC 2009*, Springer LNCS(5867):182–196, 2009.

[KH11] M. Krause and M. Hamann. The Cryptographic Power of Random Selection. *Proceedings of SAC 2011*, Springer LNCS(7118):134–150, 2011.

Efficient Compiler for Secure Two-Party Computations in ANSI C

Andreas Holzer*, Martin Franz†, Stefan Katzenbeisser**, Helmut Veit* and Nikolaos P. Karvelas**

* TU Wien † CrypTool Project ** TU Darmstadt & CASED

In his seminal paper [Yao82], Andrew Yao describes the "millionaires problem", thereby introducing the problem of Secure Two-party computation (STC): Two millionaires want to find out who is wealthier without revealing to each other their actual wealth. In it's more general form the problem can be defined as having two parties A and B, who want to compute a publicly known function f on their private inputs x_A and x_B without revealing them, while at the end they both learn the result of the evaluation. Due to it's many applications (secure electronic auctions, analysis of private data and analysis of medical signals to name a few) the problem has received a lot of attention in the cryptographic community. Implementations of the proposed STC protocols follow one of the two directions of either applying homomorphic encryption schemes, or using the solution proposed by Yao, known as "Garbled Circuits". In the latter, the function is transformed into a Boolean circuit which is encrypted in a specific way.

Although Yao's idea is simple, implementing it is not an easy task, since the transformation from function to a circuit is laborious and error prone. Attempts like [HEKM11] made significant steps towards efficient STC solutions but suffer from the fact that they offer only limited libraries, thus leaving the implementation of the circuits to the programmer, who does not necessarily have the required cryptographic background to realise such a task.

In response to the above constructions, we proposed in [HFKV12] a different approach: Using the model checker CBMC, constructed in [CKL04], we translate any C program to an equivalent Boolean circuit, which can be passed on to any of the aforementioned STC architectures. The result is a compiler for STC on any functionality written in ANSI C. In developing this tool further, our aim is to make further optimizations on the circuit sizes, allow floating point arithmetic and provide integration with other STC implementations like the one constructed in [Mal11].

References

[HFKV12] Andreas Holzer, Martin Franz, Stefan Katzenbeisser, and Helmut Veith. Secure two-party computations in ansi c. In *ACM Conference on Computer and Communications Security (CCS)*, pages 772–783. ACM Press, 2012.

[Mal11] Lior Malka. Vmcrypt: modular software architecture for scalable secure computation. In Yan Chen, George Danezis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 715–724. ACM, 2011.

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations. In *IEEE Foundations of Computer Science (FOCS)*, pages 160–164. IEEE Computer Society, 1982.

[HEKM11] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster secure two-party computation using garbled circuits. In *USENIX Security Symposium*. USENIX Association, 2011.

[CKL04] Edmund M. Clarke, Daniel Kroening, and Flavio Lerda. A tool for checking ansi-c programs. In Kurt Jensen and Andreas Podelski, editors, *TACAS*, volume 2988 of *Lecture Notes in Computer Science*, pages 168–176. Springer, 2004.

Differential Fault Analysis on Grøstl-256 and Countermeasures

Christian A. Reuter
Justus-Liebig-Universität Gießen
Arndtstraße 2
35392 Gießen
christian.a.reuter@math.uni-giessen.de

As seen in the last twenty years, it is not sufficient to prove only the theoretical security of a cryptographic algorithm. With the use of cryptographic implementations one has also to take Side Channel Attacks like Simple Power Analysis and Differential Power Analysis into account. But instead of measuring and analysing the energy consumption of the cryptographic hardware device, one induces faults. Then the faulty output is compared with the correct output – this is called Differential Fault Analysis (DFA). DFA gained a lot of attention in 1996, when Boneh, DeMillo and Lipton presented the "Bellcore"-Attack on the RSA-CRT algorithm [BC97]. With the correct and a faulty ciphertext, achieved by inducing only one fault, they retrieved the private key. A few years later the block cipher Advanced Encryption Standard (AES) was released and many DFA-attacks were developed (e.g., [Gi03]).

Now we will realize a differential fault analysis on Grøstl-256 [Grøstl], which is a cryptographic hash algorithm and was one of the five candidates for the next secure hash standard SHA-3 [NS3C]. Grøstl is based on the main structures of AES – it has a very similar algorithm structure, operates on states and uses the same substitution box as AES does. However, the classical fault attacks on AES can not be adapted directly, but give a general basis. New strategies are needed to be developed to attack Grøstl successfully.

For SHA-3, there are four slightly different versions of Grøstl; we will focus on Grøstl-256. The attack is able to completely recover the whole input message using a one-bit and a random-byte fault model. One needs less than two minutes of computation and on average 296 faults to retrieve the correct input data. Furthermore, the attack is able to yield the secret key of a keyed hash function like HMAC, with Grøstl used within.

To prevent Grøstl from being attacked by differential fault analysis, three countermeasures will be presented. One of them isn't even using more resources to compute the output of the hash algorithm. Although being successfully attacked, Grøstl still can be repaired using a very basic countermeasure.

References

[BC97] Boneh, D., DeMillo, R.A., Lipton, R.J.: On the Importance of Checking Cryptographic Protocols for Faults. In: W. Fumy (Ed.) Advances in Cryptology - EUROCRYPT '97. LNCS, vol. 1233, pp. 37-51. Springer, Heidelberg (1997)

[Gi03] Giraud, C.: DFA on AES. In: IACR Cryptology ePrint Archive, Report 2003/008 (2003)

[Grøstl] Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schlffer, M., Thomsen, S.S.: Grøstl - a SHA-3 candidate. (2011) http://www.groestl.info/Groestl.pdf

[NS3C] National Institute of Standards and Technology, Cryptographic Hash Algorithm Competition, SHA-3. http://csrc.nist.gov/groups/ST/hash/sha-3/index.html

The cube-attack

Frank-M. Quedenfeld

University of Kassel

We want to introduce the cube-attack described in [1].

Let $\mathbb{B} := \{0,1\}$ be the elements of the the field of size 2. Moreover, we have two distinct sets of variables $V := \{v_1, \ldots, v_{n_v}\}$ (IV variables) and $K := \{k_1, \ldots, k_{n_k}\}$ (key variables) for the total number of variables $n = n_v + n_k$. Based on this we define the function

$$f(V,K) : \mathbb{B}^n \to \mathbb{B}\,.$$

We also write $f(V,K)$ (or $f(K,V)$) to stress on which set of variables we work. We know that there is an algebraic normal form (ANF) for the function f. Let $M \subset \{\mu \subset (V \cup K)\}$ be a set of monomials. We then can write the ANF of f as

$$f(V,K) = \sum_{\mu \in M} \prod_{x \in \mu} x$$

over $\mathbb{B}$. We also write $\mu \in f$ instead of $\mu \in M$. By convention, we set $x_\mu := \prod_{x \in \mu} x$ and also $\prod_\emptyset := x_\emptyset := 1$. In addition, we write $x \in x_\mu$ for $x \in \mu$.
Let $C \subset V$ be a subset of the iv variables and $x_C := \prod_{x \in C} x$ a monomial. Moreover, we have

$$f(V,K) = x_C p(K) + r(V,K)$$

for $p(K)$ a polynomial solely over K and a residual polynomial r over all variables $V \cup K$. If $\nexists \mu \in r : x_C | \mu$, we call C a *cube* and x_C the corresponding *cube monomial*. In addition, $p(K)$ is called the *superpoly* of the cube C. By its definition, it gives information on the key variables of a given cipher. We call the degree $\deg(p)$ of the superpoly the *key-degree of a cube* and the number of elements $c := |C|$ in C the *dimension of the cube*. Note that this coincides with the degree of the cube monomial $\deg(x_C)$ so we have $c = \deg(x_C)$. For the k-dimensional cube C let I_C be a subset of $\mathcal{P}(V)$ including 2^k vectors in which we assign all the possible combinations of 0/1 values to variables in C. In [1] the following observation of the construction above has been proved.
Theorem. For any polynomial p and cube C it holds

$$p(K) = \sum_{v \in I_C} f(v,K).$$

So we can express a cipher as a black box polynomial f and get *superpolys* p for some cube C. This allow us to get informations about the key of the cipher. In the talk we want to explain how to get cubes and corresponding *superpolys*. Furthermore we show some generalizations from the original cube attack in [1].

Literatur

[1] A. Shamir I. Dinur. Cube attacks on tweakable black box polynomials. In *EUROCRYPT. Lecture Notes in Computer Science*, volume 5479, pages 278–299. Springer, 2009.

Combined Algebraic Side-Channel Attacks

Konstantin Böttinger

Fraunhofer Research Institution AISEC, Munich, Germany

We present work in progress of combined algebraic side-channel attacks, a new approach in cryptanalysis of hardware security tokens. Our method enables simultaneous attacks on multiple hardware implementations of cryptographic primitives at the same time in order to exploit compositional weaknesses. We generate a single combined algebraic representation of multiple primitives and show how to automatically search for malicious combinations in the token API. As a consequence, it is indicated for future architectures to consider the collectivity of implemented cryptographic primitives as a whole.

Sage als Werkzeug für Kryptographie und Kryptanalyse

Christopher Wolf

chris@Christopher-Wolf.de, Christopher.Wolf@rub.de
Fakultät für Mathematik, Horst Görtz Institut für IT-Sicherheit
Ruhr-Universität Bochum

Kryptologische Forschung findet in mehreren Stufen statt. Da ist im Teil eins Bleistift & Papier, Kreide & Tafel, die Diskussion unter Kollegen zum Finden von validen Hypothesen. Teil zwei ist die Hypothesenvalidierung bzw. Falsifizierung [Pop34]. Hier kommt zunehmend der Computer zum Einsatz, da die entsprechenden Objekte einfach zu groß bzw. zu zahlreich sind, um händisch betrachtet zu werden. Teil drei in dieser Aufzählung wären mathematische Experimente wie Faktorisierungsrekorde oder Brute-Force-Angriffe auf kryptographische Primitive. Letzte benötigen sehr große Hardware-Ressourcen, aber auch sehr schnelle Software—also C++, C, ggf. Assember.

In diesem Vortrag geht es primär um den *zweiten* Teil, also der Hyptothesenfalsifikation. Ihrer Natur nach werden hier vergleichsweise viele Hypothesen in kurzer Zeit getestet. Dabei geht es i.d.R. *nicht* darum, dies besonders effizient zu tun, sondern einfach relativ schnell Code zu schreiben, der die gewünschte Frage klärt—und dies in aktzeptabler Zeit; typisch sind 5 Minuten, eine Mittagspause oder eine Nacht. Seltener eine Woche.

Speziell für diesen Fall bietet sich *„rapid prototyping“* in einer reichen Programmiersprache an. Beispiel hierfür ist das Open Source Tool *Sage*, das praktisch alle für Kryptologie relevante Bereiche der Mathematik abdeckt; entsprechend ist es mit ca. 350 MB an reinem Source-Code auch sehr groß.

Während andere Systeme wie z.B. Magma in sich geschlossener und logischer sind, hat Sage den ultimativen Vorteil, dass es *transportabel* ist: Beim Wechsel des Arbeitgebers hat man weiterhin Zugriff auf seine Programme. Des Weiteren liegen alle Algorithmen im Sourcecode vor. So notwendig kann man sie daher anpassen. Explizit *nicht* gedacht ist Sage für Teil 3—also das Hochleistungsrechnen.

Dieser Vortrag gibt eine kurze Einführung in Sage mit Code-Beispielen.

Literatur

[Pop34] Karl Popper: Logik der Forschung, 1934.

[Sage] Open Source Mathematical Softare *SAGE*—System for Algebra and Geometry Experimentation, http://www.sagemath.org/

Towards Fresh Re-Keying with Leakage-Resilient PRFs: Cipher Design Principles and Analysis

Sonia Belaïd[1], Fabrizio De Santis[2,3], Johann Heyszl[4], Stefan Mangard[3],
Marcel Medwed[5], Jörn-Marc Schmidt[6], François-Xavier Standaert[7], Stefan Tillich[8]

[1] Ecole Normale Supérieure and Thales Communications, France.
[2] Institute for Security in Information Technologies, Technical University of Munich.
[3] Infineon Technologies AG, Neubiberg, Germany.
[4] Fraunhofer Research Institution AISEC, Munich, Germany.
[5] NXP Semiconductors, Graz, Austria.
[6] IAIK, Graz University of Technology, Austria.
[7] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium.
[8] Department of Computer Science, University of Bristol, UK.

Abstract. Leakage-resilient cryptography aims at developing new algorithms for which physical security against side-channel attacks can be formally analyzed. Following the work of Dziembowski and Pietrzak at FOCS 2008 [2], several symmetric cryptographic primitives have been investigated in this setting [1,3,5,7,8,4]. Most of them can be instantiated with a block cipher as underlying component. Such an approach naturally raises the question whether certain block ciphers are better suited for this purpose. In order to answer this question and as a preliminary step, we consider a leakage-resilient re-keying function and evaluate its security at different abstraction levels. That is, we study possible attacks exploiting specific features of the algorithmic description, hardware architecture and physical implementation of this construction. These evaluations lead to two main outcomes. First, we complement previous works on leakage-resilient cryptography and further specify the conditions under which they actually provide physical security. Second, we take advantage of our analysis to extract new design principles for block ciphers to be used in leakage-resilient primitives. That is, while our re-keying function is not (yet) a cryptographically secure block cipher, we hope that the design principles on which it is based will trigger the interest of symmetric cryptographers to design new block ciphers combining good properties for secure implementations and security against black box (mathematical) cryptanalysis.

References

1. Yevgeniy Dodis and Krzysztof Pietrzak. Leakage-resilient pseudorandom functions and side-channel attacks on feistel networks. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2010.
2. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008.
3. Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical leakage-resilient symmetric cryptography. In Prouff and Schaumont [6], pages 213–232.
4. Marcel Medwed, François-Xavier Standaert, and Antoine Joux. Towards super-exponential side-channel security with efficient leakage-resilient prfs. In Prouff and Schaumont [6], pages 193–212.
5. Krzysztof Pietrzak. A leakage-resilient mode of operation. In Antoine Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 462–482. Springer, 2009.
6. Emmanuel Prouff and Patrick Schaumont, editors. *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*. Springer, 2012.
7. François-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald. Leakage resilient cryptography in practice. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 99–134. Springer Berlin Heidelberg, 2010.
8. Yu Yu, François-Xavier Standaert, Olivier Pereira, and Moti Yung. Practical leakage-resilient pseudorandom generators. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 141–151. ACM, 2010.

Catena : A Memory-Consuming Password Scrambler

Christian Forler, Stefan Lucks and Jakob Wenzel

Bauhaus-Universität Weimar,
Germany
{Christian.Forler, Stefan.Lucks, Jakob.Wenzel}@uni-weimar.de

Passwords are user-memorizable secrets, commonly used for user authentication and cryptographic key derivation.[1] Typical (user-chosen) passwords often suffer from low entropy and can be attacked by trying out all possible password candidates in likelihood-order until the right one has been found. In some scenarios, when a password is used to open an interactive session, the security of password-based authentication and key derivation can be enhanced by dedicated cryptographic protocols defeating "off-line" password guessing, see, e.g., [1] for an early example. Otherwise, the best protection are cryptographic password scramblers, performing "key stretching". The basic idea of such schemes is using an intentionally slow one-way function for hashing the password. Therefore, the password processing take some time for both kinds of users legitimate ones and attackers.

This paper introduces Catena , a new one-way function for that purpose. Catena is sequentially memory-hard, which hinders massively parallel attacks on cheap memory-constrained hardware, such as recent "graphical processing units", GPUs. Furthermore, Catena has been designed to resist cache-timing attacks. This distinguishes Catena from scrypt, which is also sequentially memory-hard, but which we show to be vulnerable to cache-timing attacks. Additionally, Catena supports

- *client-independent updates* (the server can increase the security parameters and update the password hash without user interaction or knowing the password),
- a *server relief* protocol (saving the server's resources at the cost of the client), and
- a variant Catena-KG for secure *key derivation* (to securely generate many cryptographic keys of arbitrary lengths such that compromising some keys does not help to break other keys).

References

[1] S.M. Bellovin and M. Merrit. Encrypted key exchange: Password-based protocols secure against dictionary attacks. Proceedings of the I.E.E.E. Symposium on Research in Security and Privacy (Oakland), 1992.

[1] In our context, "passphrases" and "personal identification numbers" (PINs) are also "passwords".

On Increasing the Throughput of Stream Ciphers

Frederik Armknecht and Vasily Mikhalev

University of Mannheim
Mannheim
Germany

Important practical characteristics of a stream cipher are its throughput and its hardware size. A common hardware implementation technique for improving the throughput is pipelining where computations within the cipher are parallelized. However it requires to store intermediate values, making additional memory necessary which is the most expensive part in terms of the area size and power consumption.

For stream ciphers with feedback shift registers (FSRs), we present an alternative approach for parallelizing operations with almost no grow of the hardware size by cleverly re-using existing structures. It is based on the fact that FSRs are usually specified in Fibonacci configuration, meaning that at each clock all but one state entries are simply shifted.

Some interesting results were recently made on the methods for constructing so-called Galois-configuration NLFSRs [2, 3, 4] where the separate feedback function can be connected to each stage of a register, which can be considered as the generalisation of classical ones (of Fibonacci-configuration). The advantage of such NLFSRs is that several feedback functions can be computed in parallel which allows to generate the binary sequences faster, with no loss in security.

In this work we provide a technique how to integrate parts of the stream cipher outside of the FSR, e.g., the output function, directly into the FSRs. The technique can be seen as a combination of the two approaches mentioned above (pipelining and FSR-transformation). The idea is to parallelize the computation of the output function by integrating parts of it into several update functions of the FSR. Of course care needs to be taken that this transformation of the cipher does not alter its functionality. Thus the idea is to correct the changes made in the FSR at a later stage.

We formally describe the transformation and its preconditions and prove its correctness. Moreover, we demonstrate our technique on Grain-128 [1], one of the eSTREAM finalists with low hardware size. Our technique allows an implementation, realized by the Cadence RTL Compiler considering UMC L180 GII technology, where the throughput is increased in the initialization mode by 18% and in the keystream generation mode by 24% (compared to a time-optimized implementation without any structural changes). As opposed to other solutions, no additional memory is required. In fact the hardware size even decreased from 17876 μm^2 to 16863 μm^2.

References

[1] Martin Hell , Thomas Johansson , Er Maximov , Willi Meier. A Stream Cipher Proposal: Grain-128. *In Information Theory, 2006 IEEE International Symposium on (pp. 1614-1618). IEEE*

[2] Jean-Michel Chabloz, Shohreh Sharif Mansouri, and Elena Dubrova. An Algorithm for Constructing a Fastest Galois NLFSR generating a given sequence. In Claude Carlet and Alexander Pott, editors, *SETA*, volume 6338 of *Lecture Notes in Computer Science*, pages 41–54. Springer, 2010.

[3] Elena Dubrova. A Transformation from the Fibonacci to the Galois NLFSRs. *IEEE Transactions on Information Theory*, 55(11):5263–5271, 2009.

[4] Elena Dubrova. A Scalable Method for Constructing Galois NLFSRs with period 2^{n}-1 using Cross-Join pairs. *IACR Cryptology ePrint Archive*, 2011:632, 2011.

A family of 6-to-4-bit S-boxes with large linear branch number

Daniel Loebenberger* and Michael Nsken*

* b-it
53117 Bonn
Germany

We propose a family of 6-to-4-bit S-boxes with linear branch number 3. Since they also fulfill various further desirable properties such S-boxes can serve as a building block for various block ciphers.

It is still work-in-progress to prove that this new S-box would make DES(L) invulnerable to linear cryptanalysis. Though this may seem to be only of historical interest it sheds some light to future constructions.

A CL-Signature for Blocks of Messages and Accumulated Values

Patrick Märtens
Justus-Liebig-Universität Gießen
Arndtstraße 2
35392 Gießen
patrick.maertens@math.uni-giessen.de

Digital signature schemes are fundamental primitives in cryptographic protocols. In 2002 Camenisch and Lysyanskaya [6] constructed the first signature scheme with efficient protocols, for (1) issuing a signature on committed values, and (2) proving knowledge of a signature on committed values. This type of signature schemes, commonly referred to as *CL-signatures*, are building blocks for anonymous cryptographic systems, such as group signatures and electronic cash. In [5] Camenisch, Lysyanskaya and Hohenberger use this signature scheme to design the first *compact* e-cash scheme. Other compact and *divisible* electronic cash schemes [2, 1, 4] apply bounded accumulators [8, 2]. Hence, Au *et al.* [1] constructed a CL-signature for signing an accumulator together with a block of messages.

However, this signature scheme is only secure in type 3 pairing groups under the AWSM assumption. In this paper, we design a more efficient CL-signature for signing blocks of messages together with an accumulator, that is secure in all three pairing types.

To design this signature scheme, we combine an existing CL-signature with the accumulator scheme and a polynomial commitment scheme ([7]) and prove its security under the Strong Diffie-Hellman assumption [3].

References

[1] M. H. Au, W. Susilo, and Y. Mu. Practical Anonymous Divisible E-Cash From Bounded Accumulators, In *Financial Cryptography and Data Security*, 287–301, 2008.

[2] M. H. Au, Q. Wu, W. Susilo, and Y. Mu. Compact E-Cash from Bounded Accumulator, In *Topics in Cryptology – CT-RSA*, 178–195, 2007.

[3] D. Boneh and X. Boyen. Short Signatures Without Random Oracles, In *Advances in Cryptology – EUROCRYPT*, 56–73, 2004.

[4] S. Canard and A. Gouget. Multiple Denominations in E-cash with Compact Transaction Data, In *Financial Cryptography and Data Security*, 82–97, 2010.

[5] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact E-Cash, In *Advances in Cryptology – EUROCRYPT*, 302–321, 2005.

[6] J. Camenisch and A. Lysyanskaya. A Signature Scheme with Efficient Protocols, In *Security in Communication Networks*, 268–289, 2002.

[7] A. Kate, G. Zaverucha and I. Goldberg Constant-Size Commitments to Polynomials and Their Applications, In *Advances in Cryptology – ASIACRYPT*, 177–194, 2010.

[8] L. Nguyen. Accumulators from Bilinear Pairings and Applications to ID-based Ring Signatures and Group Membership Revocation, In *Topics in Cryptology – CT-RSA*, 275–292, 2005.

Hardware Trojan Detection: Challenges and Approaches

Nisha Jacob

Fraunhofer Research Institution AISEC, Munich, Germany

More and more manufacturers outsource the fabrication of integrated circuits (ICs) in order to reduce production costs. This separation of the design and fabrication gives third parties the opportunity to maliciously modify the manufactured IC, i.e., introduce a hardware trojan. Therefore, it is necessary to verify that the ICs are trojan-free in a post-fabrication process. Hardware trojans are known to be stealthy, making their detection during regular IC testing difficult. Researchers are therefore developing special techniques to detect hardware trojans. In this work, we give a brief introduction into the topic of hardware trojans, discuss previously proposed detection techniques, and identify the challenges for future research in this area.