Challenges in Designing Trustworthy Cryptographic Co-Processors

Ricardo Chaves, Giorgio Di Natale, Lejla Batina, Shivam Bhasin, Baris Ege, Apostolos Fournaris, Nele Mentens, Stjepan Picek, Francesco Regazzoni, Vladimir Rozic, et al.

Challenges in Designing Trustworthy Cryptographic Co-Processors

Ricardo Chaves*, Giorgio Di Natale†, Lejla Batina‡, Shivam Bhasin§‡‡, Baris Ege‡, Apostolos Fournaris††, Nele Mentens‖, Stjepan Picek‡, Francesco Regazzoni**, Vladimir Rozic‖, Nicolas Sklavos††, and Bohan Yang‖
*INESC-ID, IST, Universidade de Lisboa; †LIRMM (University of Montpellier II/CNRS); **ALaRI - University of Lugano;
‡Digital Security Group - ICIS, Radboud University Nijmegen; §TELECOM ParisTech; ‡‡Temasek Labs, NTU, Singapore;
††KNOSSOSnet Research Group, Technological Educational Institute of Western Greece; ‖KU Leuven - COSIC and iMinds;

***Abstract*—Security is becoming ubiquitous in our society. However, the vulnerability of electronic devices that implement the needed cryptographic primitives has become a major issue. This paper starts by presenting a comprehensive overview of the existing attacks to cryptography implementations. Thereafter, the state-of-the-art on some of the most critical aspects of designing cryptographic co-processors are presented. This analysis starts by considering the design of asymmetrical and symmetrical cryptographic primitives, followed by the discussion on the design and online testing of True Random Number Generation. To conclude, techniques for the detection of Hardware Trojans are also discussed.**

I. Introduction

Cryptographic and security systems are increasingly used in several critical aspects of society, namely: telecommunications, banking, commerce, government, defense, and national security. These systems provide security services such as confidentiality, integrity, and authentication. The reliability and security of these systems is the most important factor when building trust and confidence between consumers, companies, and state.

Although security systems are now a mass product, their implementation remains a challenging task, since the designer has to meet applications constrains and at the same time cope with the inherent leakage characteristics of the physical devices. This creates the challenge of designing cryptographic systems that are efficient while at the same time robust against attacks. Concerning all the possible security threats, the vulnerability of electronic devices supporting the implementation of cryptographic functions has become a major issue. Indeed, even though recent crypto-algorithms and protocols have been proven resistant to cryptanalysis, fraudulent manipulation and analysis on the platforms implementing such algorithms and protocols can allow extracting confidential information.

Given the importance of these security system in today's society and the effectiveness of the existing attacks, several research projects and coordinated activities are appearing all around the world. Among them, the COST Action IC1204 (TRUDEVICE) started in 2012 [1]. COST is an intergovernmental framework for European Cooperation in Science and Technology, allowing the coordination of nationally funded research on an European level. The TRUDEVICE Action encompass more than 150 researchers from more than 50 universities, research centers and industries, from 23 European countries. The goal of this Action is to ease research activities among different research groups in Europe, by fostering the generation of new ideas, discoveries and processes as well as the exploitation of the related results in the domain of hardware security and trust. This paper presents recent advances on some of the research topics investigated in TRUDEVICE. In particular, herein we focus on asymmetric and symmetric cryptographic hardware design with Side-Channel Attack (SCA) resistance, True Random Number Generator (TRNG) testing and the existing Hardware Trojan detection techniques.

This paper is structured as follows: Section II presents an overview of the most significant attacks targeting the implementation on cryptographic processors, given their usage and manufacturing procedures. Sections III and IV present a more detailed discussion focused on the design and implementation of efficient and side-channel resistant asymmetric and symmetric algorithms. Section V provides an overview on embedded TRNG testing while Section VI elaborates on Hardware Trojans and the existing detection techniques.

II. Overview of Existing Attacks to Cryptografic Implementations

The possible attacks that a security device can be subjected to depends on the life cycle of that same device. Considering the life-cycle of the cryptographic device, a wide range of attacks need to be considered. These range from the semiconductor manufacture up until its deactivation, passing through the usage phase. This is particularly valid, since the attacker can be the legitimate user of the device, such as in set top boxes or ID cards.

The device can be attacked during its fabrication phase. This attack consist in the insertion of malicious circuits, that can have multiple purposes, ranging from the leakage of sensitive information to the complete malfunctioning of the system [2]. These malicious circuits are designated as Hardware Trojans. Detection of Hardware Trojans involves the deployment of several techniques in order to isolate the infected circuits from good ones, such as logical testing, side-channel analysis, or reverse engineering.

During the initialization phase, as well as during the standard operation of the device, random and equiprobable values need to be produced. From these values unpredictable and reliable keys can be generated for encryption and authentication,

978-1-4799-8391-9/15/$31.00 ©2015 IEEE

of data and of the device itself. Key generation is an essential aspect of the system, since if these values are manipulated or predicted to some extent, the security of the entire system can be compromised. To derive good cryptographic keys TRNGs are needed [3]. However, the generated numbers are not necessarily statistically perfect, since some bias and correlation between the bits might be present. Additionally, the output of a TRNG can be compromised by a hardware failure or manipulated by an active attack.

Alternatively, unique and device dependent values can also be generated by Physical Unclonable Function (PUF). PUFs are dedicated circuit primitives that allow to generate values from the unique and complex physical characteristics of the Integrated Circuit (IC). Since the values can be re-generated by the PUF, each time they are needed, they do not need to be stored in memory, thus preventing their leakage under invasive or fault injection attacks [4]. However, PUFs cannot be accurately mathematical modeled and it is hard to properly evaluate their correct behavior [5]. Given the dependency on the circuit characteristics, the output exhibits variations each time it is read. This implies the need for error corrections codes, that need to stored in memory. Additionally, if not properly implemented PUF circuits may leak information and be susceptible to penetration attacks, voiding their main characteristics [6].

During the operation phase of the device, the other main target of the attacker is the computation of the cryptographic algorithms themselves, in particular symmetric and asymmetric encryption algorithms. During this phase, the device can be subjected to several additional attacks such as fault injection attacks, side-channel attacks, and physical tampering. While physical tempering attacks tend to be destructive and very expensive, Fault attacks (FA) and SCA can be much less or even non invasive while being cheaper and easy to perform.

Fault Attacks are based on the intentional modification of the circuit's environment (e.g., applying extreme temperature, exposing the IC to radiation, X-rays, ultra-violet or visible light, or tampering with the clock frequency) in such a way that the function implemented by the device generates an erroneous result. The attacker can then discover secret information by comparing the erroneous result with the correct one. In-the-field detection of any failing behavior is therefore of prime interest towards taking further actions, such as discontinuing operation or triggering an alarm [7].

Side-channel attacks take advantage of unintentional physical leakages such as timing, power dissipation, electromagnetic radiation, etc. The best known example of SCA is Differential Power Analysis (DPA) [8]. This attack exploits the dependence between the power consumed (in a hardware circuit) an the secret data being processed. The adversary uses these observations to recover the secret keys. Other physical leakage, such as timing [9] or electromagnetic emanations [10], can be exploited for the same purpose. Both, symmetric and asymmetric cryptographic implementations are vulnerable to these attacks. SCA can be improve if combined with theoretical cryptanalysis, leading to the so-called algebraic SCA [11].

III. Efficient And Attack Resistant Asymmetric Cryptographic Primitives

Standardized asymmetric key cryptographic systems are associated to the Discrete Logarithm problem (DLP), the Elliptic Curve (EC) DLP and the number factorization problem resulting in schemes like RSA or El Gamal and EC cryptographic (ECC) schemes [12]. Cryptographic engineering on asymmetric cryptography aims at optimizing the performance of the most computationally intensive operations (modular exponentiation in RSA/El Gamal or Scalar multiplication in ECC) and on the implementation of accelerators robust against physical attacks (targeting SCA or FA resistance).

Asymmetric key cryptosystems are based on cyclic group arithmetic operations (Z_p : RSA / El Gamal, GF(p)/$GF(2^k)$ fields: ECC) that are optimized in terms of hardware resources (memory, chip covered area) and speed (time delay). These optimizations are focused on modular multiplication using Montgomery modular multiplication, Karatsuba-Offman or Barret's reduction algorithms as well as alternative number representations like Residue Number System.

However, these optimizations may cause the circuit more susceptible to SCA and FA. These attacks aim at the Asymmetric key cryptosystem primitives like modular multiplication. SCAs on these primitives are thwarted either by implementing additional circuitry to make leakage trace constant or/and by realizing SCA resistant algorithms when computing the results. Such variants like the Montgomery Power Ladder, square-and-always-multiply/double-and-always-add or Binary Random Initial Point (BRIP) algorithm [13] provide protection against simple SCAs but not against more sophisticated SCAs including refined Power Attacks (PAs), zero PAs and comparative simple PAs (doubling and relatively doubling attacks) [13] as well as DPAs. The above attacks can be thwarted through randomization/masking and hiding [14]. FAs are also very powerful attacks against asymmetric cryptography primitives and associated mathematical parameters (in the case of ECC) especially when combined with SCAs. By injecting faults during a cryptographic computation flow, FAs can create cryptographic weaknesses (in the case of ECC transforming the EC into a cryptographically weak one) and/or faulty results that can reveal the key. FA countermeasures are based on the infective computation principle with appropriate fault detection mechanisms and randomization as well as hardware, fault tolerance based techniques like dual rail circuitry [15].

IV. Improving side-channel resistance of S-boxes

One of the most sensitive part of a secure symmetric cryptographic algorithm considering SCA is the substitution operation, often implemented using the so-called S-Boxes, the widely used non-linear building blocks in block ciphers. When evaluating S-boxes, researchers need to consider a vast number of different security properties. Each of the properties characterize the resistance of an S-box against a certain attack.

In addition, when considering the SCA resistance of S-boxes, the situation is even more complex. The main reason is in a contradicting criteria for ensuring both, the SCA

resistance and theoretically secure S-boxes. In particular, the non-linearity property implies more theoretically secure substitution operations, while at the same time weaker against side-channel attacks, ciphers.

Starting from the initial work of Kocher et al. [8], the topic of differential power analysis receives an undivided interest from both academia and industry. This fact is largely related to the practicality of these attacks and as a consequence this topic also attracts some attention from cryptographic algorithm designers. When block ciphers are concerned, the resistance to attacks of a cryptographic algorithm, such as differential [16] and linear [17] cryptanalysis, are well studied. However, improving the resistance to DPA together with improved resistance against linear and differential attacks are shown to be a contradicting phenomena [18].

In literature, there are several metrics proposed to quantify the resistance of a block cipher against power analysis. In 2004, Guilley and Pacalet proposed SNR (DPA) as a first attempt to quantify the level of leakage expected from a design under certain assumptions [19]. Shortly after that, Prouff proposed the "transparency order" in an attempt to compare S-boxes in terms of their resistance to DPA [20]. In 2012 Fei et al. take another approach and propose the "confusion coefficient" for S-boxes which quantifies how distinguishable two key candidates can be in the case of DPA[21].

All those efforts suggest a clear interest in this type of research but we are still far from taking a measure for SCA resistance into consideration when designing a new cipher. However, there is a clear potential with this type of research. Namely, improving the SCA security intrinsically could decrease the costs of SCA countermeasures, which is a big problem for low-cost applications such as RFIDs, sensor nodes, etc.

V. Embedded tests for TRNG

To properly evaluate the quality of TRNG, embedded tests are needed in order to properly monitor the noise source. Towards this, the German standard AIS-31 [22] proposes three classes of embedded tests: start-up tests after power-up, the tot-test to detect a total failure, and continuous tests which are slower, but capable of detecting more subtle statistical weaknesses.

Usual TRNGs consist of a noise source, an entropy extractor and a post-processing block. The noise source is the component that generates all the randomness in the system, sometimes in the form of an analog signal. The entropy extractor converts this signal into raw random numbers. These numbers are not necessarily statistically perfect, since some bias and correlation between the bits might be present. To cope with this, post-processing is used to compress the raw bits into full-entropy internal bits. Additionally, embedded tests can also be applied, in particular at three specific points: after the entropy source (analog signal), after the extractor (on raw bits) or after the post processing block (on internal bits). The closer the testing to the entropy source, the more reliable it is.

An example of direct monitoring of the entropy source is embedded jitter measurement to ensure that enough jitter is generated. This solution was implemented in [23].

Raw bits and internal bits can be monitored using statistical tests. The common idea behind the statistical tests is to measure a property of the generated sequence (such as the bias or the frequency of appearance of a given pattern) and to calculate the probability that an ideal RNG produces a sequence that is worse than the measured one. If this probability is below a given threshold, the test fails. FPGA implementations of selected statistical tests from FIPS and NIST test suites are provided in [24] and [25]. An ASIC implementation of 6 selected NIST tests is presented in [26]. The common limitation of the reported statistical tests implementations is the fact that they work only for Independent Identically Distributed (IID) values. If an entropy source providing non-IID values is used with the post-processing block, raw bits will always fail these tests. To the best of our knowledge there are no implementations of embedded statistical tests that would be suitable for non-IID entropy sources.

VI. Hardware Trojans Detection Techniques

Detection of Hardware Trojans at an early stage is extremely important because, unlike software, hardware Trojan cannot be removed once inserted. Since the nature of Trojans varies widely, it is not possible to develop a unique detection technique. The state of the art literature on Trojan detection can be divided into two wings, *viz.*: *destructive* and *non-destructive*.

Destructive methods exploit techniques like reverse engineering to detect Trojans. Sophisticated, expensive, and highly accurate techniques like Scanning Optical Microscopy (SOM) etc. are deployed to reconstruct the design layout and eventually the original netlist. Such techniques can become impractical when applied on a large number of ICs. Recently [27] authors present a new approach to detect Trojans by visual inspection which studies the cross-correlation between images of the last metal layer directly obtained from the manufactured die to the ones produced by design tools before manufacturing.

The non-destructive methods can be further classified as: *invasive* and *non-invasive*. Non-invasive techniques for Trojan detection compares the performance characteristics of the target circuit with a "golden model". A detection mechanism called DEsign-For-ENabling-SEcurity (DEFENSE [28]) adds reconfigurable logic to the functional design in order to implement security monitoring at real time. Some testing techniques could also be deployed to detect Trojans. Standard testing techniques may not be effective for Trojan detection owing to their extremely low activation probability. Banga et al. [29] use the inverted output of flip-flops $\overline{Q}$ in order to raise the control over them and enlarge the space of reachable states. Jha and Jha [30] propose a randomization technique to probabilistically compare the functionality of the original design and the final circuit. Tehranipoor et al. [31] presented a method to increase the probability of generating a transition in a Trojan and analyze its activation time. All these techniques cannot entirely ensure the triggering of Trojans and therefore

its detection as the test patterns are very complex and design dependant.

Another popular technique to detect Trojans in ICs is SCA. One of the first work by Agarwal et al. [32] proposed to use of Principle Component Analysis (PCA) for extracting a side-channel fingerprint of an IC and to compare it to the one of the golden model. Further physical characteristics which can be used to detect Trojans are leakage current, dynamic current, or internal delays.

VII. Conclusion

This paper highlights the challenges in designing trustworthy cryptographic systems, with an emphasis on the existing attacks to their implementations. Particularly, this paper discusses how computational optimizations can open a window to side-channel and fault attacks or how an attempt to reduce leakage may lead to mathematically weaker encryption algorithms. Additionally, we survey the generation of true random numbers and online testing of the generated values. To conclude, the existence of possible Hardware Trojans and their detection is analyzed. Overall, this paper gives some directions to the needed research on the design and implementation of trustworthy cryptographic systems.

Acknowledgments

This work was partially supported by the TRUDEVICE COST action (ref. IC1204), the ARTEMIS Joint Undertaking under grant agreement n. 621429 and UID/CEC/50021/2013.

References

[1] G. Di Natale, "TRUDEVICE: A COST Action on Trustworthy Manufacturing and Utilization of Secure Devices," *Information Security Journal: A Global Perspective*, vol. 22, no. 5-6, pp. 205–207, 2013.

[2] U. O. Defense, "Defense science board task force on high performance microchip supply," *Washington, DC*, pp. 2005–02, 2005.

[3] V. Fischer, "A Closer Look at Security in Random Number Generators Design," in *COSADE*, 2012, pp. 167–182.

[4] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proceedings of the 44th annual Design Automation Conference*. ACM, 2007, pp. 9–14.

[5] S. Katzenbeisser, Ü. Kocabaş, V. Rožić, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, "PUFs: Myth, fact or busted? A security evaluation of physically unclonable functions (PUFs) cast in silicon," in *Cryptographic Hardware and Embedded Systems–CHES 2012*. Springer, 2012, pp. 283–301.

[6] D. Nedospasov, J.-P. Seifert, C. Helfmeier, and C. Boit, "Invasive PUF analysis," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on*. IEEE, 2013, pp. 30–38.

[7] H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan, "The sorcerer's apprentice guide to fault attacks," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 370–382, 2006.

[8] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," in *Advances in Cryptology - CRYPTO'99*, ser. LNCS, no. 1666. Springer-Verlag, 1999, pp. 388–397.

[9] D. A. Osvik, A. Shamir, and E. Tromer, "Cache Attacks and Countermeasures: The Case of AES," in *Proceedings of the 2006 The Cryptographers' Track at the RSA Conference on Topics in Cryptology*, ser. CT-RSA'06, 2006, pp. 1–20.

[10] J.-J. Quisquater and D. Samyde, "ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smard Cards," in *Smart Card Programming and Security (E-smart 2001)*, ser. LNCS, vol. 2140. Springer-Verlag, 2001, pp. 200–210.

[11] M. Renauld, F.-X. Standaert, and N. Veyrat-Charvillon, "Algebraic side-channel attacks on the AES: Why time also matters in DPA," in *Cryptographic Hardware and Embedded Systems-CHES 2009*. Springer, 2009, pp. 97–111.

[12] A. J. Menezes, P. C. V. Oorschot, S. A. Vanstone, R. L. Rivest, and S. A. V. Alfred J. Menezes Paul C. Van Oorschot, "Handbook of Applied Cryptography," 2001.

[13] J. Fan and I. Verbauwhede, "An Updated Survey on Secure ECC Implementations: Attacks, Countermeasures and Cost," in *Cryptography and Security: From Theory to Applications*, ser. Lecture Notes in Computer Science, D. Naccache, Ed. Springer Berlin Heidelberg, 2012, vol. 6805, pp. 265–282.

[14] J.-S. Coron, "Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems," in *Proceedings of the First International Workshop on Cryptographic Hardware and Embedded Systems*, ser. CHES 1999. London, UK: Springer-Verlag, 1999, pp. 292–302.

[15] M. Joye and M. Tunstall, *Fault Analysis in Cryptography*. Springer, 2012.

[16] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," in *Proceedings of CRYPTO '90*, ser. LNCS. London, UK: Springer-Verlag, 1991, pp. 2–21.

[17] M. Matsui and A. Yamagishi, "A new method for known plaintext attack of FEAL cipher," in *Proceedings of EUROCRYPT'92*, ser. LNCS. Berlin, Heidelberg: Springer-Verlag, 1993, pp. 81–91.

[18] C. Carlet, "On highly nonlinear S-boxes and their inability to thwart DPA attacks," in *Proceedings of INDOCRYPT'05*, ser. LNCS. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 49–62.

[19] S. Guilley and R. Pacalet, "Differential Power Analysis Model and Some Results," in *Proceedings of CARDIS 2004*. Kluwer Academic Publishers, 2004, pp. 127–142.

[20] E. Prouff, "DPA attacks and S-boxes," in *Fast Software Encryption*. Springer, 2005, pp. 424–441.

[21] Y. Fei, Q. Luo, and A. A. Ding, "A Statistical Model for DPA with Novel Algorithmic Confusion Analysis," in *Proceedings of the 14th International Workshop on Cryptographic Hardware and Embedded Systems CHES 2012*, ser. LNCS, 2012, pp. 233–250.

[22] W. Killmann and W. Schindler, "A Proposal for: Functionality classes for random number generators ," ser. BDI, Bonn, 2011.

[23] V. Fischer and D. Lubicz, "Embedded Evaluation of Randomness in Oscillator Based Elementary TRNG," in *CHES*, 2014, pp. 527–543.

[24] R. Santoro, O. Sentieys, and S. Roy, "On-line Monitoring of Random Number Generators for Embedded Security," in *ISCAS*. IEEE, 2009, pp. 3050–3053.

[25] F. Veljković, V. Rožić, and I. Verbauwhede, "Low-cost implementations of on-the-fly tests for random number generators," in *DATE*, 2012, pp. 959–964.

[26] V. B. Suresh, D. Antonioli, and W. P. Burleson, "On-chip lightweight implementation of reduced NIST randomness test suite," in *HOST*, 2013, pp. 93–98.

[27] S. Bhasin, J.-L. Danger, S. Guilley, X. T. Ngo, and L. Sauvage, "Hardware Trojan horses in cryptographic IP cores," in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on*. IEEE, 2013, pp. 15–29.

[28] M. Abramovici and P. Bradley, "Integrated circuit security: new threats and solutions," in *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*. ACM, 2009, p. 55.

[29] M. Banga and M. S. Hsiao, "Odette: A non-scan design-for-test methodology for trojan detection in ics," in *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 18–23.

[30] S. Jha, "Randomization based probabilistic approach to detect trojan circuits," in *High Assurance Systems Engineering Symposium, 2008. HASE 2008. 11th IEEE*. IEEE, 2008, pp. 117–124.

[31] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A novel technique for improving hardware trojan detection and reducing trojan activation time," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 112–125, 2012.

[32] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan detection using IC fingerprinting," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. IEEE, 2007, pp. 296–310.