Hacking and protecting IC hardware

Saïd Hamdiaoui, Jean-Luc Danger, Giorgio Di Natale, Fethulah Smailbegovic, Gerard van Battum, Mark Tehranipoor

Hacking and Protecting IC Hardware

Said Hamdioui
Computer Engineering
Delft University of Technology, the Netherlands

Giorgio Di Natale
Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, France

Gerard van Battum
Brightsight
The Netherlands

Jean-Luc Danger
Secure IC
France

Fethulah Smailbegovic
ESCRYPT GmbH – Embedded Security
Bochum, Germany

Mark Tehranipoor
TrueLogic & Uni of Connecticut
USA

***Abstract*—Traditionally most of people treat a hardware solution as an inherently trusted box. "it is hardware not software; so it is secure and trustworthy", they say. Recent research shows the need to re-asses this trust in hardware and even in its supply chain. For example, attacks are performed on ICs to retrieve secret information such as cryptographic keys. Moreover, backdoors can be inserted into electronic designs and allow for silent intruders into the system. And, even protecting intellectual-property is becoming a serious concern in the modern globalized, horizontal semiconductor business model. This paper discusses hardware security, both from hacking and protecting aspects. A classification of all possible hardware attacks is provided and most popular attacks are discussed including the countermeasures.**

Keywords— Site-channel attacks, Hardware Trojans, fault injection, counterfeiting.

I. INTRODUCTION

Since the invention of the first integrated circuit (IC) in 1958 and introduction of first standalone Central Processing Unit (CPU) in 1971, we witnessed and continue to observe the breathtaking advances in IC manufacturing, transistor density and architectural solutions. These advances fueled the imagination of developers so that we now have diverse application fields for integrated circuits; from RF ID chips and microcontrollers to CPUs for desktop PCs with billion transistors integrated. ICs and systems have become a multibillion-dollar business and represent the physical backbone of our digitalized world. Interesting enough, they are being increasingly deployed even in many security-critical infrastructures such as sensitive governmental organizations, military, and financial/banking systems, where the impact and consequences of *attacks* could be catastrophic. Till recently, we have intuitively trusted the chips to control our lives and processes, so we have huge amount of sensitive information processed in chips. However, nowadays, attacks are being launched increasingly for economic reasons by well-funded criminal organizations or for intelligence purposes to get access to secret and sensitive information. Moreover, the emergence of globalized and horizontal IC and semiconductor business model, mainly driven by cost savings, is requiring both designs and users re-asses their trust in hardware and even in the supply chain. In recent years many reports have appointed to these attacks on the electronic components and their supply chain [1]. The semiconductor industry is today loosing over $4 billion a year due to these kind of attacks; not to mention the catastrophic results these attacks could have for critical applications [2,3,4,5].

Depending on their targets, hardware attacks can be classified into three classes:

- ***IC data (assets) attacks***: These are attacks that aim at retrieving the secret data of the IC; e.g., hacking a smart cart to get the secret key;
- ***IC design (IP) attacks***: These are attacks that aim at getting more information on the IC design in order to counterfeit it; e.g., perform reverse engineering on an IC or IP, steal and/or even claim the ownership;
- ***IC functionality (tampering) attacks***: these are attacks that target the alternation of the original function of the chip/system. For example, a chip ceases functioning or continues to operate but then in an impaired manner, a chip introducing corruption in the data, etc.

In this paper we will focus on the first two classes. Most known attacks within each class will be described and the means of avoiding them will be discussed. In addition future challenges in hardware security will be highlighted.

II. HACKING ICS FOR DATA

This section provides first a taxonomy and a classification of the different types of hacking ICs for data. Thereafter the most important three types will be discussed in details.

A. Classification

Depending on either they cause the chip and/or the packing to be damaged or not, attacks can be further divided into three categories [6,7] as shown in Figure 1:

- *Invasive attacks*: These are attacks requiring direct access to the internal of the device and therefore that do harm the chip and destroy its packaging; they typically require high skills and specialize laboratory. They are typically very time consuming, ranging from hours to weeks. Therefore, they are expensive.
- *Non-invasive attacks*: These do not physically damage the chip. They require moderately sophisticated equipment and are typically low cost as compared with invasive attacks. Obviously they are more dangerous than invasive ones as the owner of the device will never notice that his device is hacked.

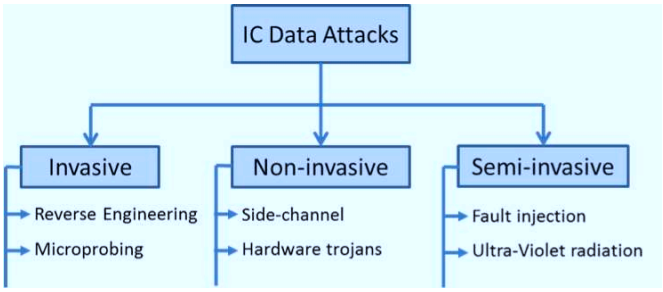

Figure 1: IC data attacks classification

- *Semi-invasive attacks*: These do require de-packaging the chip in order to get access to its surface, as it is the case for invasive attacks. However, the passivation layer of the chip remains intact, as semi-invasive methods do not require depassivation or creating contacts to the internal lines. They fill the gap between the first two categories, being both inexpensive and easy repeatable.

Figure 1 reports also some known attacks in the literature for each type. From all of these, three are most used for IC data hacking; these are side-channel, hardware trojans, and fault injection. They are discussed next.

B. Side-channel attacks (SCA)

Every chip has observable physical properties or circuit activity such as power consumption, heat, electromagnetic radiation or time to complete an operation. The information gained from these physical properties can be used to extract information from the chip. In [8] an attack was presented which used side-channel information to attack cryptographic hardware. This attack measured the execution time of cryptographic operations to determine parts of the cipherkey. This type of attack based on observations is called simple side-channel analysis. The complex side-channel analysis is based on statistical techniques that combine multiple measurements to extract the secrets. Over the years, it has been discovered that power consumption, as side-channel information, is far more effective. Today we have successful Simple and Differential Power Analysis (SPA and DPA) as methods of attack using information about chip power consumption.

It is worth noting that these attacks correspond to the ones of the "poor", as they are not invasive (hence do not require costly equipment and skills) and just require the knowledge of the algorithm and how to access the device. However, they need a minimum knowledge of the implementation and could require some competences in signal acquisition, digital signal processing and statistics. Indeed, the computation time and the energy consumed by the computer are indirectly linked to the secret information, which physically "leaks" to the external world.

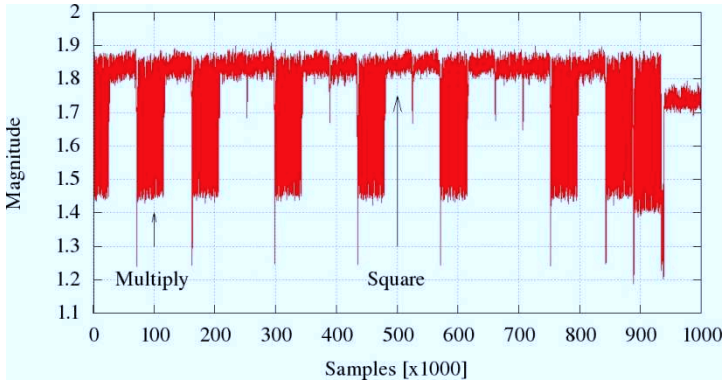

Figure 2: Simple Power Analysis

As an example, consider the exponentiation operation of the RSA cryptographic algorithm. When this algorithm is executed, the "Simple Power Analysis" (**SPA**) can easily distinguish the measured activity of "square" , (corresponding to exponent bit at '0') from those of "square and multiply", (corresponding to exponent bit at '1') as shown in the example of Figure 2. This activity is measured by means of an oscilloscope and a current or electromagnetic (EM) probe. The figure illustrates the trace of an RSA exponentiation activity measured from an EM probe located on top of the device. It is clearly possible to extract the exponent bits from the observed pattern when differentiating the square from multiply patterns.

More powerful SCA take advantage of statistical properties, as the CPA for "Correlation Power Analysis". CPA is very efficient to attack private key cryptography like DES or AES, which are naturally robust against SPA. The attack principle is to compare the activity observation with a predictor of sensitive variables. These attacks are led with a "divide-and conquer" approach where the secret is unveiled piece by piece. The difficulty of the attacker is to find the best predictor based on the activity of a sensitive variable that depends on the secret information. Generally the digital CMOS technologies activity depends on the variable value (mathematically modelled as Hamming weight of a vector variable), or the transitions of this variable (Hamming distance).

C. Hardware Trojans

The process of design and implementation of chips became fine granular over the years. In the frontend design we do not design everything, but buy some or the most of design blocks (intellectual property IP cores) from specialized third parties. Also, physical implementation, manufacturing and packaging are done by different companies nowadays. At each of these steps of chips supply chain, there is a possibility of malicious alteration of hardware. This malicious alteration of hardware (also called Hardware Trojan) by the attacker can result, under specific conditions, in opening the access to data on the chip. We know little about the real world of Hardware Trojans because there are no reported incidents involving Hardware Trojans (yet), but we accumulate research results related to their creation and detection. Nevertheless, it is clear that the

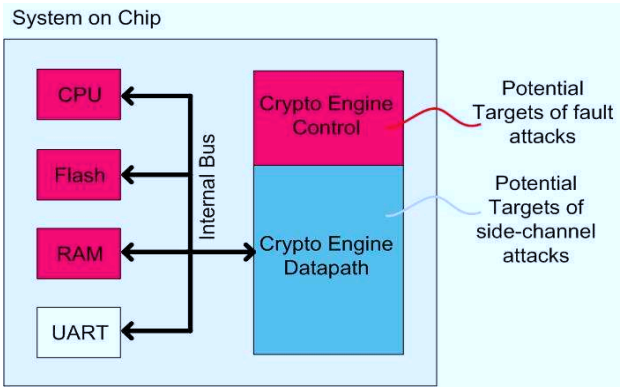

Figure 2: Combining two attacks

globalized economy and advances in chip complexity make Hardware Trojans an increasingly possible scenario.

Hardware Trojans can be classified into functional and parametric [9]; the functional class is realized by adding or deleting transistors or gates and the parametric class is realized by modifying existing wires, transistors and logic. For example, in [10], the functional class of hardware Trojans is demonstrated. The attacker is able to control the system and get unlimited access to memory by inserting the Hardware Trojan into the CPU. In [11], the parametric class of Hardware Trojans is demonstrated on Intel's Random Number Generator used in Ivy Bridge processors. This example of Trojans does not need extra logic resources but requires only a change in dopant polarity of a few transistors.

Hardware Trojans can be further classified according to their activation or action characteristics. Hardware Trojans can be activated internally or externally and as a consequence of activation can transmit stolen information, modify specification or chip function [12].

D. Fault injection attacks

Another powerful hardware attack is the "Fault Injection Attack" (FIA). Faults can be natural and induced by the environment in which the chip operates; examples are radiation, electrical noise and overheating, which may result in chip malfunction. However, faults can be deliberately injected into the chip by an external attacker interested in learning about the chip and extracting sensitive and secret information and Differential Fault Analysis (DFA) has been proven to be a powerful tool, since only a handful of faulty ciphertexts are needed to extract the secret key (cf. [13] and [14]). For instance, the fault can flip a control bit value to disable a protection or algorithms and take advantage of comparisons between correct and faulty results.

The FIA attack is typically invasive as the fault has to be injected; e.g., by using laser equipment, hence needing devices to be unpackaged. Recent fault attacks take advantage of EM injections, which make them less invasive but also provides less accuracy about the targeted computation block where the secret is involved [15].

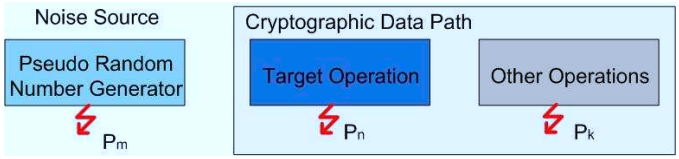

Figure 1: : Increasing resistance to attacks by adding high noise level

E. Combinations of attacks

The methods of attacking the chips evolve and will evolve in the future, and so will the countermeasures. As the complexity of chips will continue to grow, so will the complexity of attacks, where two or more standard attacks can be combined. For example, the basic idea, presented in [16], is to combine both the side channel attack (observation) with the fault injection attack (perturbation) as shown in Figure 3 [17]. When combining different attacks, the probability of hacking the chip increases as it needs at least one of the two methods to succeed. Note that in [16,17], the concept combined attacks relies on the fact that fault injection countermeasures often react at the end of execution, making the opening for side-channel attack (power analysis).

III. PREVENTING IC DATA ATTACKS

This section reviews some of the countermeasures against the discussed IC data attacks in the previous section. Of course it will be ideal to have countermeasures that make it impossible to hack an IC using any other attack, while having minimum or no impact on area overhead and performance. However, apparently there are no ideal models to prevent the success of an attack. The quality of a countermeasure is typically measured in the effort required for a successful attack given a certain platform.

A. Countermeasures against side-channel attacks

The goal of countermeasures against side-channel attacks is to implement the chip (e.g., crypto hardware) in such way that the attacker's effort in retrieving the sensitive information is too high to be continued and successfully completed. Countermeasures can be implemented at different levels of design and implementation, including circuit/gate [18,19,20] and micro-architectural levels [21,22].

For instance, the following ways can be used to implement micro-architectural countermeasures against side-channel attack based on power analysis [21].

- *Adding Noise*: By adding a Pseudo Random Number Generator (PRNG), extra noise is added to the power measurements; see Figure 4. The higher the noise, the higher the number of measurements required for a successful attack, hence the higher the resistance to the attack.
- *Dummy Operations*: In a DPA (Differential Power Analysis) attack, the attacker observes power consumption of the same operations in large number of measurements. If the continuity of the observed operation

can be interrupted, than the attacker would be forced to collect much more data. The interruption is done by adding dummy operations.

- *Alternative Logic Styles*: a DPA attack can be effectively countered if the power consumption is made independent for the data processing. Alternative logic styles are proposed, like asynchronous logic or dual-rail pre-charge logic style.
- *Masking*: To counter a DPA attack, there are attempts to solve this at algorithmic level. This countermeasure prohibits direct operations between key and data by adding random mask to data prior to cryptographic operations. If for each run of a DPA a different mask for data is used, then the DPA attack will be effectively prevented.
- *Design Methodology*: a Globally Asynchronous Locally Synchronous (GALS) System with different asynchronous clocks. The design is partitioned into islands of logic with different clocks. Clocks are present in the power measurements, but the attacker cannot easily attribute a given clock signal to the correct island.

B. Countermeasures against hardawre trojans

In their presence in an IC, and irrespectively where they were injected (pre-manufacturing and the post-manufacturing phase), Hardware Trojans have to be detected either at pre-manufacturing and/or the post-manufacturing phase to prevent the effected hardware from being integrated in the system/application.

Detection can happen in the pre-manufacturing and the post-manufacturing phase [11]. In the pre-manufacturing phase, the detection is based on the completeness of chip verification. However, if a (potentially untrustworthy) third party supplier of IP blocks is involved, additional logic can be added between their IPs to make Trojan activation more difficult [23]. Moreover, by using unique chip properties/features, hardware Trojans can be also detected at the design stage. For instance, in [24] the authentication of hardware by checking its implementation at low level has been demonstrated. The microarchitecture features of the chip are complex and unique such that a unique checksum can be computed; this checksum is based on a cycle-to-cycle activity of the microarchitecture and it has been shown that small differences can result in significant deviations in the checksum; hence detecting malicious alteration of hardware.

For detection of the Trojans at the post-manufacturing phase, the "golden chip" approach can be used. A golden chip is known to be free from hardware Trojans and is used for comparison to other chips of the same functionality. Here both reverse engineering (e.g., use the scanning electron microscope to make photos of all layers of the chip and compare them to the layout masks in order to detect additions to layers or wires) or side-channel information (e.g., collect the of the golden chip information on power, electromagnetics, or time and compare with the that of the chip under investigation) can be used as means of Trojan detections [25, 26, 27].

HW Trojans design, analysis, implementation and detection are topics for further research. Even though there are no reported incidents involving hardware Trojans, we have already accumulated research that could help us in fighting this type of security problems. Globalized IC business model and advances in chip complexity make hardware Trojans an easy-feasible scenario.

C. Countermeasures against fault injection

The countermeasures against fault injection (perturbation) attacks can be classified into four classes [28]:

- *Integrity Check for Inputs*: many fault injection attacks tries to (a) exploit forcing the computation to take place in a different way that originally implemented, or (b) exploit properties of some chosen inputs. Checking the unexpected properties on inputs can prevent such attacks [29].
- *Parallel Redundant Computations*: algorithms can be extended with redundancy to detect manipulations [30, 31, 32, 33].
- *Inherent Algorithm Properties*: some algorithms already have an inherent type of redundancy, and checking them can help in detecting the faults [34,35].
- Sensors: built-in transient error detector (based for instance on the bulk current sensors) can be used to trigger an alarm whenever a possible attach is detected [36].

IV. IC DESIGN ATTACKS

As already mentioned, IC design (IP) attacks aim at getting more information on design in order to *counterfeit* it. As the complexity of the electronic systems and integrated circuits increased significantly over the past few decades, they are mostly fabricated and assembled globally to reduce the production cost. This globalization has led to an illicit market willing to undercut the competition with counterfeit and fake parts.

In the rest of this section we first briefly propose a taxonomy of counterfeit type. Thereafter IC supply chain vulnerability will be discussed; and finally the countermeasure will be described.

A. Counterfeit components

As defined in [2], a counterfeit component has one of the following properties: (i) is an unauthorized copy; (ii) does not conform to original component manufacturer (OCM) design, model, and/or performance standards; (iii) is not produced by the OCM or is produced by unauthorized contractors; (iv) is an off- specification, defective, or used OCM product sold as "new" or working; or (v) has incorrect or false markings and/or documentation.

Based on the definition above and analyzing supply chain vulnerabilities, we classify the counterfeit types into seven

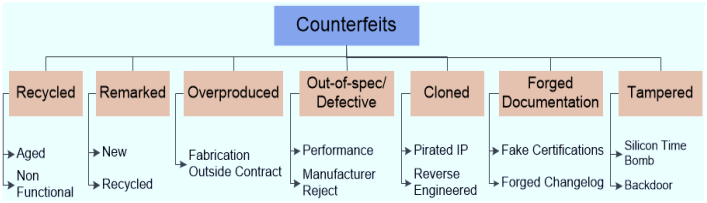

Figure 5: Taxonomy of counterfeit types

distinct categories shown in Figure 5. *Recycled* refers to an electronic component that is reclaimed/recovered from a system and then modified to be misrepresented as a new component of an OCM. The *remarking* is accomplished by either chemically or physically removing the original marking, blacktopping (resurfacing) the surface to hide any scratches or imperfections that have been created, and then remarking the new surface. *Overproduction* occurs when foundries and packaging companies sell components outside of contract with the design house (component's intellectual property (IP) owner). A part is considered *defective/out-of-spec* if it produces an incorrect response to post-manufacturing tests. These parts should be destroyed, downgraded, or otherwise properly disposed of. *Cloning* is widely used by a range of adversaries/counterfeiters (from small entities to large organizations) to copy a design in order to eliminate the large development cost of a part. Some fake parts may be supplied with *forged documents*. Finally, some parts may be *tampered* with malicious inclusion.

B. *Supply chain vulnerability*

Typically an electronic component will go through a process as shown in Figure 6. This process includes design, fabrication, assembly, distribution, usage in the system, and finally end of life. There are vulnerabilities associated with each step of the process. Attacks on the *design* stage can be performed in the two following ways: (i) the counterfeiter can steal the intellectual properties (IPs) to create cloned components, (ii) the counterfeiter can tamper with codes to modify the functionality, create backdoors, etc. An untrusted *foundry* can potentially (i) make extra/overproduced ICs, by hiding their yield, and selling those extra ICs in the open market, (ii) clone the design, and (iii) source defective and out-of-specification wafers to packaging companies to make finished parts. An untrusted *assembly* can (i) build overproduced ICs by hiding the yield information, (ii) sell the defective/out-of-specification ICs, and (iii) remark, forge, or upgrade a component's marking. There are two types of *distributors* – authorized and unauthorized – in the supply chain. The threat lies mostly from unauthorized distributors. There are several reports pointing to phony distributors potentially sourcing all seven types of counterfeit components in the supply chain. An untrusted *system integrator* can potentially use all types of counterfeit components in their system. They can maximize the profit by using the cheap or tampered counterfeit components. When electronics age or become outdated (*end-of-life*), they are typically retired/resigned and subsequently replaced. Proper disposal

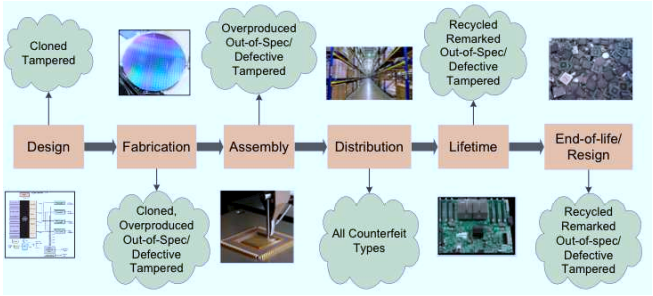

Figure 6: Electronic components supply chain vulnerabilities

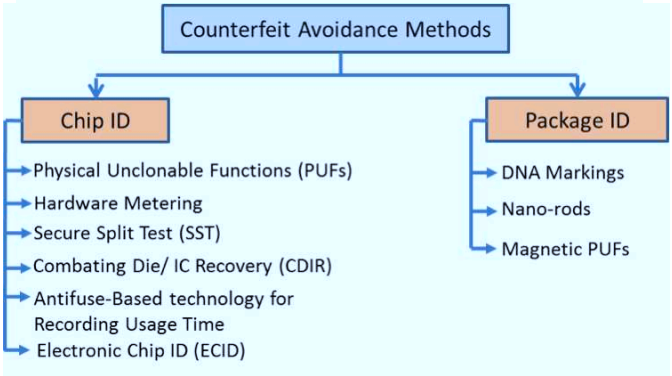

Figure 7: A taxonomy of counterfeit avoidance techniques

techniques are highly advised to ex- tract precious metals and to prevent hazardous materials (lead, chromium, mercury, etc.) from harming the environment.

C. *Avoidence and mesaures*

Different types of components, namely obsolete, active, and new impact differently for implementing counterfeit avoidance measures. New mechanisms can be put in place during the design of new chips that could help prevent counterfeiting. As obsolete parts are no longer being manufactured, and active parts are being fabricated based on a previous design and developed masks, the focus should be on the implementation of avoidance measures at the package level. Figure 7 shows the taxonomy for such counterfeit avoidance measures. It is broadly classified into two major categories – chip ID and package ID.

The chip IDs are those inserted into the circuits. Physical unclonable functions (PUFs) can generate unique IDs for each chip given process variations [37] helping protect against cloning. Hardware metering techniques attempts to control the chip access mechanism by the foundry where the IP owner allows only a limited number of keys entered into the chip before test [38]. Secure Split-Test (SST) allows the IP owner take full control of the test process [39]. Using SST, only the chips that have passed the test would be shipped to the market. Combating die/IC recycling (CDIR) sensors take advantage of the aging in the chip to identify a recycled IC [40]. Anti-fuse based technology [41] can be used to detect chip usage in the field. Similarly, this technology can help detect recycled ICs

very effectively. Finally, electronic IDs (ECIDs) have been commonly used by semiconductor industry for field return analysis. Such technology can also protect clones and remarking of ICs.

V. FUTURE CHALLENGES

As the semiconductor industry continues progress towards smaller and smaller nodes and new microarchitectures are emerging, we witness increasing richness of applications and at the same time increasing complexity of data processing. It is difficult, and may be economically not affordable to capture every potential use case of the chip (including security) at design time and verify design before the production. Security solutions for the chips will be then incomplete; nevertheless, they should be able deal with the possible "unexpected" in field. Globalized economy with continuous pressure on time-to-market and cheaper products also shape the security solutions in chips of the future. "Too much" or "too few" security in the chips, due to incomplete design and implementation process and market forces, will help the new classes of attacks on the chips to emerge; some will be combination of different attacks, containing passive and active attacks.

When securing the chip, today designers consider different aspects like protection of inputs, processing and memory parts and the control flow. Designers follow proactive strategy of protecting chips; they anticipate the attacks and build the mechanisms to defend the chips. Designers assume that attackers are reasonable and act according to certain probability distributions. The main question is either this will work for future chips? Obviously, much research is to be done; understanding hardware security problems in the future will strongly depend on novel applications and microarchitectures. However, the following can be stressed:

- With rising complexity of chips, the complexity of defenses will also rise and probability that defenses are inadequate against some attacks. The complexity prevents us to patch every last vulnerability in the chips.
- If we could patch every last vulnerability in chips, we would invest the resources in fortification of the chip protecting the chip against attacks that may never happen. The bigger the chip is, the bigger fortification will be, and consequently the more costly the chip will be.
- We cannot assume that attackers in the future will be reasonable and act according to fixed probability distribution, as we assume about attacker today. We must make worst-case assumptions, including that attackers have knowledge of chip defenses and that all chip vulnerabilities are not patched.

So, what is the strategy for defending the chip in the future? Since we cannot patch every last vulnerability and anticipate every new attack or combination of attacks, we still have to enable the chip to react to vulnerabilities and attacks and apply defenses where they are needed. This reactive strategy with inherent flexibility may cost less than the full fortification of the chip and may respond better to previously not anticipated attacks.

VI. CONCLUSION

This paper has presented different aspects if hardware security; a classification is of all existing hardware attacks is provide. Most popular hacking methods and their countermeasures are discussed.

Hardware security and attack prevention are becoming very important aspects of today's electronics especially when considering the presence of professional well-funded (criminal) organization with the purpose hardware hacking!

REFERENCES

[1] trust-HUB, http://trust-hub.org/home.

[2] U.S. Senate Committee on Armed Services, "Inquiry into Counterfeit Electronic Parts in the Department Of Defense Supply Chain," May 2012.

[3] U.S. Senate Committee on Armed Services, "Suspect Counterfeit Electronic Parts Can Be Found on Internet Purchasing Platforms," February 2012. [Online]. Available: http://www.gao.gov/assets/590/588736.pdf

[4] Karen Mercedes Goertzel, Booz Allen Hamilton, Integrated Circuit Security Threats and Hardware Assurance Countermeasures, Cross Talk, Nov-Dec 2013.

[5] S. Mangard, Keeping Secrets on Low-Cost Chips, IEEE Security and Privacy, Vol. 11, No 4, pp. 75-77, Jul-Aug 2013.

[6] M. Tehranipoor, C. Wang, Introduction to Hardware Security and Trusts, Spring, 2012.

[7] S. Skorobogatov, Semi-invasive attacks- a new approach to hardware security analysis, Technical report UCAM-CL-TR630, University of Cambride, April 2005.

[8] Paul C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", Lecture Notes in Computer Science 1109, pp. 104–113, 1996.

[9] Mohammad Tehranipoor, Farinaz Koushanfar. "A Survey of Hardware Trojan Taxonomy". In IEEE Design&Test of Computers, , IEEE CS Press , pp.10-25, January/February 2010.

[10] S.T. King, J. Tucek, A. Cozzie, C. Grier, W. Jiang, and Y. Zhou. "Designing and implementing malicious hardware". In Proceedings of the 1st USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET 08), pages 1-8, 2008.

[11] Georg T.Becker, Francesco Regazzoni, Christof Paar and Wayne Burleson. "Stealthy Dopant-Level Hardware Trojans" Workshop on Cryptographic Hardware and Embedded Systems, CHES 2013, Santa Barbara, USA, August 20-23, 2013.

[12] X. Wang, M. Tehranipoor, and J. Plusquellic, ''Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions,'' Proc. IEEE Int'l Workshop Hardware- Oriented Security and Trust (HOST 08), IEEE CS Press, pp. 15-19, 2008.

[13] E. Biham and A. Shamir. "Differential fault analysis of secret key cryptosystems". In Burton S. Kaliski Jr., editor, Advances in Cryptology - CRYPTO '97, volume 1294 of Lecture Notes in Computer Science, pages 513–525. Springer, 1997.

[14] Christophe Clavier. "Passive and Active Combined Attacks on AES - Combining Fault Attacks and Side Channel Analysis", Workshop on Fault Diagnosis and Tolerance in Cryptography, 2010.

[15] Frederic Amiel, Karine Villegas "Passive and Active Combined Attacks – Combining Fault Attacks and Side Channel Analysis" Workshop on Fault Diagnosis and Tolerance in Cryptography, 2007.

[16] Guilley, S.; Sauvage, L.; Danger, J.-L.; Selmane, N., "Fault Injection Resilience," Fault Diagnosis and Tolerance in Cryptography (FDTC), 2010 Workshop on , vol., no., pp.51,65, 21-21 Aug. 2010.

[17] H. gebotys, et al., C. G Security Wrappers and Power Analysis for SoC Technologies, ACM/IEEE ISSS-CODES, pp. 162-167, 2003.

[18] A. Khatibzadeh, C. Gebotys, Enhanced Current-Balanced Logic (ECBL): An Area Efficient Solution to Secure Smart Cards against Differential Power Attack, Fourth International Conference on Information Technology, pp. 898-899, 2007.

[19] K. Tiri, I. Verbauwhede, A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation, Design, Automation and Test in Europe Conference and Exhibition, pp. 246-251, 2004.

[20] Gürkaynak, Frank Kağan , "GALS system design: side channel attack secure cryptographic accelerators", PhD Thesis, ETH Zürich, 2006.

[21] G. P. Hancke, Noisy Carrier Modulation for HF RFID, Proceedings of First International EURASIP Workshop on RFID Technology, pp 63–66, 2007.

[22] A. Waksman and S. Sethumadhavan. "Silencing hardware backdoors" in IEEE Symposium on Security and Privacy (SP 2011), pages 49-63, 2011.

[23] G.E. Suh, D. Deng, and A. Chan, ''Hardware Authentication Leveraging Performance Limits in Detailed Simulations and Emulations,'' Proc. 46th Design Automation Conf. (DAC 09), ACM Press, pp. 682-687, 2009.

[24] 1. D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar. Trojan Detection using IC Fingerprinting. In IEEE Symposium on Security and Privacy (SP 2007), pages 296-310, 2007.

[25] J. Li and J. Lach. At-speed delay characterization for IC authentication and Trojan horse detection. In IEEE International Workshop on Hardware-Oriented Security and Trust (HOST 2008), pages 8-14, 2008.

[26] J. Yier and Y. Makris. "Hardware Trojan detection using path delay fingerprint" in IEEE International Workshop on Hardware-Oriented Security and Trust (HOST 2008), pages 51-57, 2008.

[27] I. Verbauwhede, D. Karaklajic and J-M Schmid, "The Fault Attack Jungle - A Classification Model to Guide You", Workshop on Fault Diagnosis and Tolerance in Cryptography, pp. 3-8, 2011.

[28] M. Ciet and M. Joye, 'Elliptic Curve Cryptosystems in the Presence of Permanent and Transient Faults', Designs, Codes and Cryptography, V36, pp. 33–43, 2005.

[29] N. Ebeid and R. Lambert. Securing the Elliptic Curve Montgomery Ladder against Fault Attacks. In Fault Diagnosis and Tolerance in Cryptography (FDTC), 2009 Workshop on, pages 46 –50, sept. 2009.

[30] K.J. Kulikowski, Zhen Wang, and M.G. Karpovsky. Comparative Analysis of Robust Fault Attack Resistant Architectures for Public and Private Cryptosystems. In Fault Diagnosis and Tolerance in Cryptography, 2008. FDTC '08. 5th Workshop on, pages 41 –50, aug. 2008.

[31] E. Ozturk, G. Gaubatz, and B. Sunar. Tate Pairing with Strong Fault Resiliency. In Fault Diagnosis and Tolerance in Cryptography, 2007. FDTC 2007. Workshop on, pages 103 –111, sept. 2007.

[32] G. Di Natale, M. Doulcier, M. L. Flottes, B. Rouzeyre, A Reliable Architecture for Parallel Implementations of the Advanced Encryption Standard, Journal of Electronic Testing (JETTA), Springer, Volume 25 Issue 4-5, pp. 269-278, August 2009.

[33] A. Dominguez-Oviedo. On Fault-based Attacks and Countermeasures for Elliptic Curve Cryptosystems. PhD thesis, University of Waterloo, Canada, 2008.

[34] D. Karaklaj´c and, Junfeng Fan, J.-M. Schmidt, and I. Verbauwhede. Low-cost fault detection method for ECC using Montgomery powering ladder. In Design, Automation Test in Europe Conference Exhibition (DATE), 2011, pages 1 –6, march 2011.

[35] R. Possamai Bastos, F. Sill Torres, G. Di Natale, M. Flottes, B. Rouzeyre, Novel Transient-Fault Detection Circuit Featuring Enhanced Bulk Built-in Current Sensor with Low-Power Sleep Mode, Microelectronics Reliability (Elsevier), Volume 52, Issues 9-10, Pages 1781-1786, September-October 2012.

[36] G. E. Suh and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation," in Proc. 44th ACM/IEEE Design Automation Conf. DAC 07, pp. 9-14, June 2007.

[37] F. Koushanfar and G. Qu, "Hardware metering," in Proc. Design Automation Conference, pp. 490–493, 2001.

[38] G. Contreras, T. Rahman, and M. Tehranipoor, "Secure split-test for preventing ic piracy by untrusted foundry and assembly," in Int. Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), 2013.

[39] X. Zhang, N. Tuzzio, and M. Tehranipoor, "Identification of recovered ics using fingerprints from a light-weight on-chip sensor," in Proc. of IEEE on Design Automation Conference, pp. 703 –708, June 2012.

[40] X. Zhang and M. Tehranipoor, "Design of On-chip Light-weight Sensors for Effective Detection of Recycled ICs," IEEE Transactions on VLSI (TVLSI), 2013.