Report from Dagstuhl Seminar 16342

Foundations of Secure Scaling

Edited by

Lejla Batina[1], Swarup Bhunia[2], and Patrick Schaumont[3]

1 **Radboud University Nijmegen, NL,** lejla@cs.ru.nl
2 **University of Florida – Gainesville, US,** swarup@ece.ufl.edu
3 **Virginia Polytechnic Institute – Blacksburg, US,** schaum@vt.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 16342 "Foundations of Secure Scaling". This seminar hosted researchers in secure electronic system design, spanning all abstraction levels from cryptographic engineering over chip design to system integration. We recognize that scaling is a fundamental force present at every abstraction level in electronic system design. While scaling is generally thought of as beneficial to the resulting implementations, this does not hold for secure electronic design. Indeed, the relations between scaling and the resulting security are poorly understood. This seminar facilitated the discussion between security experts at different abstraction levels in order to uncover the links between scaling and the resulting security.

Seminar August 21–26, 2016 – http://www.dagstuhl.de/16342
1998 ACM Subject Classification Hardware, Security/Cryptology, Verification/Logic
Keywords and phrases Cryptographic Engineering, Very Large Scale Integration, Secure Hardware Design, Technology Scaling, Complexity Scaling, Secure Evaluation
Digital Object Identifier 10.4230/DagRep.6.8.65

1 Executive Summary

Lejla Batina
Swarup Bhunia
Patrick Schaumont

In electronic system design, scaling is a fundamental force present at every abstraction level. Over time, chip feature sizes shrink; the length of cryptographic keys and the complexity of cryptographic algorithms grows; and the number of components integrated in a chip increases. While scaling is generally thought of as beneficial to the resulting implementations, this does not hold for secure electronic design. Larger and faster chips, for example, are not necessarily more secure. Indeed, the relations between scaling and the resulting security are poorly understood. This Dagstuhl Seminar hosted researchers in secure electronic system design, spanning all abstraction levels from cryptographic engineering over chip design to system integration.

Foundations of Secure Scaling, *Dagstuhl Reports*, Vol. 6, Issue 8, pp. 65–90
Editors: Lejla Batina, Swarup Bhunia, and Patrick Schaumont
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Discussion Topics

The mechanisms of secure scaling require investigation of the links between Cryptography, Technology, and Digital Integration. Cryptographers are concerned with novel and secure algorithms that remain secure even as cryptanalytic capabilities improve. Technologists are concerned with the next generation of transistors and their implementation into a reliable and stable process technology. Integrators are concerned with electronic design automation tools that can manage the rapidly increasing complexity of electronic design, and the are concerned with the integration of components on a complex system-on-chip.

Through its participants, the seminar offered a unique opportunity to discuss cross-cutting topics in Secure Scaling. The following list are examples of such cross-cutting topics.

- Scaling effects in Privacy and Security. The massive amount of connected devices will create significant challenges towards security and privacy. Major questions involve data ownership and key ownership and management.
- Power/Energy Efficient Crypto: Secure wireless devices and Secure RFID are two well known examples of applications that require security under severe power and/or energy constraints. Optimizing a cryptographic algorithm for power/energy efficiency needs to consider all abstraction levels of design.
- High-Performance Crypto: Information Technology is increasingly asymmetric, with larger, high-performance servers at one end, and a large population of tiny devices at the other side. Cryptographic designs must scale towards high-performance, high-throughput implementations while it must also accommodate small-footprint, low-latency designs.
- Secure Test: Complex chips utilize a number of testing strategies such as BIST and JTAG. When a chip includes a secure part, the test infrastructure carries a potential risk of abuse. Secure Test is a test strategy for complex chips that takes this risk fully into account.
- Complexity Management in Secure SoC: Managing and integrating a secure module into system-on-chip context is challenging and creates a hard verification problem that cuts through multiple traditional layers of design. Furthermore, managing multiple stakeholders in a single chip design is extremely challenging and may result in conflicting design requirements.
- Implementation Attacks: In modern cryptographic designs, side-channel analysis, fault-analysis and physical tampering are an integral part of the threat model. This requires design techniques that fully integrate countermeasures as part of the design process. In addition, the design of a countermeasure effective against most forms of tampering is an open research issue.
- Technology effects on implementation attacks. Better insight the internal operation of secure implementations at all abstraction levels leads to novel implementation attacks, that work at finer granularity, and that use novel source of leakage such as optical leakage.

The seminar supported participants in learning about the state-of-the-art developments in the three different domains covered in the workshop (Cryptography, Integration, and Technology). The seminar also supported the presentation of specific cross-cutting topics, as well as round-table (panel-style) discussions.

2 Table of Contents

Table 1 Schedule of talks.

Day	Monday	Tuesday	Wednesday	Thursday	Friday
Topic	**Cryptography**	**Integration**	**Technology**	**Cross-Cutting**	
Chair	Lejla Batina	Swarup Bhunia	Ingrid Verbauwhede		
Speaker	Bart Preneel	Sandip Ray	Christian Boit	Guido Bertoni	Joan Daemen
	KU Leuven, BE	NXP, US	TU Berlin, DE	ST MicroElectronics, IT	ST Microelectronics, BE
Topic	IoT Privacy and Security	Secure SoC Design	Optical Interaction	Glitches and SCA	Smart Card Protocol
Speaker	Kazuo Sakiyama	Swarup Bhunia	Shahin Tajik	Svetla Nikova,	Patrick Schaumont
	UEC, JP	University of Florida, US	TU Berlin, DE	KU Leuven, BE	Virginia Tech, US
Topic	Double Arbiter PUF	Trojan Detection	Secure FPGA	Threshold Implemen.	The Stovepipe Model
Speaker	Thomas Eisenbarth	Yier Jin	Naofumi Homma	Francesco Regazzoni	
	WPI, US	U of Central Floriday, US	Tohoku University, JP	Alari, CH	
Topic	Side-channel Analysis	IoT Security Metric	EM Side-channels	Design Scaling	
Speaker	Roy D Basu	Peter Schwabe	Nele Mentens	Georg Sigl	
	IIT Kharagpur, IN	Radboud University, NL	KU Leuven, BE	TU Munich, DE	
Topic	Lightweight ECC	Timing Side-channels	Dynamic Reconfig.	Atatck Scaling	
Speaker		Bilgiday Yuce		Debdeep Mukhopadhyay	
		Virginia Tech, US		IIT Kharagpur, IN	
Topic		CPU FI Attacks		Physical Security	

3 Organization

During the first three day of the seminar, discussions highlighted each major design abstraction level, and its connection to security. In the next two days, we discussed cross-cutting issues related to secure scaling. Table 1 illustrates the schedule of talks over the three days.

After each set of talks, we organized a roundtable discussion to further elaborate on discussions raised during the presentations. To keep track of the talks, we maintained a wiki that collected all slides. The presentations are available on the Dagstuhl Wiki and the organizers will encourage the participants to publicly release their slides. We also used a note-taker for each talk, who kept track of the presentation and the questions. The note-takers supported the development of this report.

4 Overview of Talks

4.1 Privacy and Security Challenges of the Internet of Things

Bart Preneel (KU Leuven, BE)

Notes taken by Joan Daemen.

The number of electronic devices connected to the internet has been growing exponentially and will continue to do so in the coming years. This is often called the Internet of Things (IoT). Many of these devices process and transmit personal information giving rise to privacy concerns. Additionally, the interconnectivity allows remotely monitoring and controlling things like medical sensors (some even implanted), cars (soon self-driving), aeroplanes, industrial infrastructures and power plants giving rise to security concerns. Moreover, during the last decades a number of organizations such as NSA, Google, Facebook have started using this infrastructure for mass surveillance for varying reasons including financial profit. Nowadays, we see a vast and complex ecosystem of companies trading in privacy-sensitive information of citizens for advertizing. Powerful data analysis techniques (Big Data) are applied to these data giving rise to an alarming level of knowledge present in this data.

Moreover, advances in genome decoding (and coding) technology will allow the commercial and political exploitation of our most personal data: our DNA. The large complexity of these ecosystems and the pace by which this evolves has lead to a lack of legal regulation. Moreover, the little regulation that there is, is not enforced. Citizens are stimulated to manage their personal data on remote servers owned and managed by corporate organizations. This is often referred to as the cloud.

There is no commonly agreed definition of privacy: it differs a lot per country and culture. Still, it is clear that from the privacy and security point of view, the situation is alarming and getting worse. Doom scenario's where evil forces abuse the information in the cloud for power abuse, terrorism or even world domination are not far fetched. Moreover, these systems have not been designed with security, safety or privacy in mind. Due to the complexity and evolving nature bugs and malware are introduced at a faster pace than their detection. Often products and apps are rolled out without any security protection and the idea is to add security later. In most cases, there is no incentive for implementing good security as the business model is often that the cost of fraud/breakdown is charged to the end customer. For some popular products such as Skype and Whatsapp we are given the impression that confidentiality is guaranteed by end to end encryption, but there is no good way to verify this and for Skype even evidence of the opposite. The Snowden revelations were an eye-opener on the amount and pervasiveness of the surveillance but after some initial indignation the world went back to business as usual.

The question relevant to this workshop is now: what can be our role in this as cryptographers and system designers and implementers? Our work is often used to protect the interest of the mentioned conglomerates rather than the interest of citizens. This is not a simple technical question but more of what we want our society to be and for that we should involve sociologists. But of course in any case the technical challenge of adding some security to these systems is formidable. The development cycle of hardware has become so complex with so many parties involved that the presence of trojans (hardware or software) is not improbable. What we can do is concentrate of sub-parts and try to do a good job there. Simplicity, open source and transparency are good guidelines in that. Education on security is also a good investment.

4.2 Double Arbiter PUF and Security Evaluation Using Deep Learning

Kazuo Sakiyama (The University of Electro-Communications, JP)

Notes taken by Nele Mentens.

- Introduction
- Previous work of the presenter
  - Fault Sensitivity Analysis: fault analysis not requiring the ciphertext
  - RFID tag:
    - complete analog/digital chip, largest crypto building block was Keccak, mutual authentication possible at a distance of 10 cm
    - no difference in timing with or without Keccak because the analog part is dominant!
- Contributions of today's talk about PUFs:

- improvement of arbiter PUF (double arbiter PUF - DAPUF)
- Q-Class authentication (instead of using only 2 classes, so 0 or 1)
- using deep learning to evaluate the security

- Related work:
  - arbiter PUF -> ML attack -> N-XOR PUF -> double arbiter PUF -> ML attack on N-XOR PUF
- DAPUF details + results:
  - based on the idea of WDDL with complementary logic
  - can be extended to 3-1 and 4-1 DAPUF by XORing all comparisons
  - comparison of 2-XOR PUF and 2-1 DAPUF:
    - uniqueness is improved compared to APUF
    - randomness is not that good
  - comparison of 3-XOR PUF and 3-1 DAPUF:
    - uniqueness and randomness good
    - steadiness becomes more random for 3-1 and 4-1 DAPUF
- Q-Class authentication:
  - steadiness more random is good for ML resistance but not good for reliable authentication, that's why Q-class authentication is used, introducing multiple response classes
  - e.g. for some challenges all the responses are 0 (class 1), for some challenges all the responses are 1 (class 4), for some challenges a number of responses are 1 and a number of responses are 0 (class 2 and 3 depending on the percentage of 0/1)
  - 64 consecutive identical challenges are applied
  - the response to the verifier is the class number
  - experimental setting: use deep learning to build a clone and measure the responses, more secure if we go to 3-1 and 4-1 DAPUFs
- Q&A:
  - How to make sure that a fair comparison can be done between the two parts? Copy the lay-out + effort is made to make routing to the XOR balanced
  - Why XOR? XOR is chosen to introduce noise, because the FPGA does not generate enough noise
  - How is the training done for the ML attack? The class numbers are used
  - Why are the percentages of the 4 classes not exactly balanced? It turned out better for the ML attack resistance

4.3 IoT and Implementation Security

Thomas Eisenbarth (Worcester Polytechnic Institute, US)

Notes taken by Georg Sigl.

Security and privacy are a major concern for the IoT: Computing is everywhere using as well as generating sensitive data. IoT creates tough power, size and cost constraints. Furthermore the attack surface increases due to the physical accessibility of the IoT devices. Spoofing of sensors will impact the physical world leading to safety and other risks. Designers have to find an optimum of cost, performance, and security. This leads to new solutions

like lightweight crypto and authenticated encryption. The main challenges on IoT are the communication interfaces, 30 years lifetime of devices, and physical attacks. The smart card industry is aware of physical attacks and has developed a certification process, which deals with that. This process is however very slow. Fail safe implementation techniques against SCA could speed up development and tools could be developed to apply them. Standardized tests like T-test or MIA should be used for security verification. Threshold implementations are a good standardized countermeasure which can be easily implemented even for lightweight crypto like Simon. How can we detect leakages in implementations? T-Test is a simple method developed by CRI. Thomas has improved the T-Test towards a paired T-Test, which eliminates common noise in pairs of measurements. But not only the IoT devices can be attacked. Attacks may be performed also against the cloud servers. An example are microarchitectural attacks: modern architectures introduce data dependent execution times due to performance optimizations. Cache attacks executed over the net will be a major threat for cloud systems.

Discussion

1. Discussion about the triangle security-cost-performance. There are other factors at system level which have to be taken into account. Examples are power consumption, latency requirements, or design time.
2. Is there a need for lightweight crypto? The use cases for lightweight crypto are very rare and may even become less if technology shrinks further. Only very low power applications with narrow communication distance can take advantage of lightweight implementations. The state cannot be reduced very much. Only implementation "tricks" help to reduce the area further. Another use case for lightweight crypto might be low latency. We currently have no good solutions for low latency cyphers in real time systems or for fast memory access.
3. What is the required randomness for threshold implementations? The required randomness is significant. Usually the area of the random number generators is not included in the area numbers. For Thomas' Simon implementation the area requirement for the random number generation will be probably the same as for the crypto implementation. The generation of randomness for all kind of SCA protected implementations is not investigated sufficiently.
4. Do TI implementations help also against other attacks? Currently TI is dedicated to SCA only. Other more algorithmic countermeasures may be better against other attacks.

4.4 A minimalistic perspective on Public Key Encryptions

Roy Debapriya Basu (Indian Institute of Technology – Kharagpur, IN)

Notes taken by Bilgiday Yuce.

Elliptic Curve Cryptography (ECC) is a promising alternative to RSA for resource-constrained systems. A lightweight (72 slices on a Spartan6 FPGA) and side-channel resistant ECC processor is proposed. The processor achieves low area by using One Instruction Set Computing (OISC) and utilizing the hardware macros on the FPGA. The processor uses one

instruction, SBN, to carry out most of the ECC operations. The execution time of Right Shift, Field Multiplication, and Shifting Key Register operations are not practically affordable when they are implemented with SBN instruction. Therefore, the processor includes low-area hardware accelerators for these operations. The processor achieves side-channel resistance with a low-cost operand swapping technique (IACR Report 925/2015).

Discussion

1. The processor has 5 variants of the SBN instruction, and it uses a 4-bit flag to choose between these variants. What would happen if we selected an undefined value of the 4-bit flag? What operation would be executed?
2. What are the basic motivations for OISC?
3. What is the area of OISC in comparison to NiosII, PicoBlaze, and MicroBlaze?
4. Are FPGAs suitable for IoT? Are not they more costly in comparison to ASIC?

4.5 Secure System Design in the IoT Regime

Sandip Ray (NXP Semiconductors – Austin, US)

Notes taken by Yier Jin.

Intel starts the project titled "Secure, Intelligent, and Reliable Internet-of-Things (StaRT)". The key idea is how we can develop smart, reliable, trustworthy systems and applications with billions of (untrustworthy, potentially malicious) computing devices. The talk is related to this project.

We are (will be) in an IoT regime. A toy IoT example is a bad coffee detector. A general IoT hierarchical structure is presented where too many configurations of sensors, devices and gateways are available. From the system design perspective, every layer should have the computation capability so that the data does not need to travel all the way from end nodes to the cloud.

There are different stakeholders of the IoT landscape but their view/goals are not quite consistent for various reasons including lacking the standard. IoT market trends are introduced among them security is important. But security is not a stand-alone standard but within other metrics.

Assets in a smartphone is introduced. The attack surface of a smartphone is also introduced. Then the solution called Platform Security Assurance is introduced. One aspect is the linking between assets and accessing policies. For a more complexity fabrics example, to build a secure architecture requires the knowledge across all layers. A review to the security architecture reveals the complexity of such architectures.

The topic then moves to the unique features of IoT, and then to the post-quantum crypto. How can we configure the IoT so that it is resilient to post-quantum attacks in, say, 20 30 years.

Discussion

1. *Ingrid*: Coffee machine case. Local vs. cloud computing. *Sandip*: It depends on the data size to select powerful platform.

2. *Swarup/Patrick*: What we should focus at this moment? Shouldn't we handle the problem as a whole problem instead of individual problems? Why is it different from a traditional SoC design procedure? *Sandip*: Divide the problem into independent problems. Traditional metrics have specific guidelines. Things started changing since the smartphone where uncertainty raises. We are not sure the IP's specific role when designing the SoC for IoT since the functionalities vary.

4.6 Where Security Meets Verification: From Microchip to Medicine

Swarup Bhunia (University of Florida – Gainesville, US)

Notes taken by Guido Bertoni.

A retrospective view: 1966 Apollo guidance computer versus today smartphone. In 50 years more powerful computers enabling new applications. From silicon micro to silicon nano electronics, next non-silicon technology.

Would the security be different from what we have in silicon techno? Could new techno provide better properties? And similar, better energy efficiency, reliability?

Question: Will the computer be a system of switches? Not known, there might be other paradigm New devices might have higher variation, good for TRNG or PUF. CMOS variation is usually very close to Gaussian, new techno might have other distributions. This could have an impact on the PUF construction.

Nano device could be very good to build memory but might have asymmetric access. Implication to side channel attack. Many new challenges and opportunity. Attacks on HW, from IC to IoT.

Define context of HW and the associated design flow. From design spec to IC fabrication, PCB fabrication to final customer. Many different points where an attack might take place. How to verify the design process?

Question: Are there evidence of inserted Trojan? Not official, there are some reports describing military chips with unexpected backdoors. Supply chain is mostly uncontrolled. PiRA, Puf in a package. Authentication method based on the resistance of PINs, like those in the GPIO.

Questions on how use cases and protocols. The physical method should be considered as a unique fingerprint of the device. Physical attack: modchip for game consoles. How to authenticate the PCB has not being modified?

Proposal: leverage JTAG in the field to check what JTAG sees in term of PCB and thus authenticate that PCB has not being altered. PCB integrity validation.

Microchip Trust Verification: recap on HW Trojan timeline. Bugs vs Malicious changes. Trust verification is quite different from traditional verification. Bigger challenge. Psot production trust validation, could be destructive but only a fraction of chips might have Trojan so not useful. Non destructive, for test all of them. Statistical test (paper CHES 2009). Use of golden chip and adopt measurement of max freq and absorbed current.

New field: food and medicine counterfeiting detecting.

4.7 Security Metric for IoT

Yier Jin (University of Central Florida, US)

Notes taken by Thomas Eisenbarth.

Summary of Talk

Ethical hacking is still hacking, i.e. at best in a legal gray area

Audience comment: "That depends on the local legislation (e.g. jail-breaking a phone is legal in europe but not in the US)." Iot and CPS describe commercial vs. industrial (or Industrie 4.0) application but are technically the same thing.

Security aspects of them include:

- "Inconvenience level:" fridge sends spam
- "Privacy:" "big personal data" might be a problem in some scenarios (pacemaker)
- "Safety issue:" car IT is hackes
- "National security issue:" critical infrastructure

Examples from literature:

- physical devices (generator, a part of the infrastructure) is blown up remotely via network attack;
- a remotely controlled jeep drives into a ditch;
- a smart gun which fires at the wrong target;

Yier has also found many vulnerabilities for IoT devices:

- Nest Thermostat: jtag debug port was not shut down
- Smart home system: design is using DES, which can be brute-forced
- Smart band: in debug mode, security stack can be disabled.
- Roku device:
- F-Secure router can be upgraded to premium device, since HW and firmware are the same
- Smart Meter: remotely read data; replaces meter with fake one; interestingly, by replacing the smart part of the smart meter, the tamper evidence feature is also 'hacked'

Take away:

- Device level hacking is a problem!
- Hackers can have a lot of patience: Reverse engineering is an option;

Solutions for IoT Security:

- Attack-oriented protections: check against a known list of attacks e.g. at: http://www.hardwaresecurity.org/iot/database/
- attacks sorted by level of system exploited
- Also provides a set of rules
- This is work in progress: other contributors welcome

Discussion

Q: What is TrapX? Attack or defense?

One device goes rogue and turns into WiFi router, forwards challenges and can monitor all communications.

Q: Hacking as described is like penetration testing? Do companies ask for this as a service?

Yes, they do. They also have bounty programs sometimes.

Q: Are any devices designed with security in mind?

They have security, but they do not properly implement, or they do not protect all levels. Example: firmware signing is becoming common, but secure boot is not: so one can replace verification code. Or they leave other obvious side channel, often simple ones.

Q: Do you follow a systematic approach for attack?

Yes, it is on the web site under Hands on Lab (see link above), but it is work in progress.

4.8 Eliminating timing side-channels in cryptographic software

Peter Schwabe (Radboud University Nijmegen, NL)

Notes taken by Swarup Bhunia.

Peter Schwabe presented his research findings on timing attacks in software, which focus on exploiting timing variations that depend on secret data. His attack models included cache attacks. He highlighted that timing attacks are serious concerns and among side-channel attacks this is only attack that can be remotely mounted on a hardware through a network. He used the square multiply example – a common operation in cryptography – to explain software timing attacks and its countermeasures through judicious low-cost software modifications. He highlighted that to prevent timing attacks, software need to follow only two rules.

1. Don't branch on secret data.
2. Don't access secret memory address.

He explained with examples how following these two simple rules through appropriate modification of a code can mitigate timing channels. For example a branch can be converted to arithmetic expression, which in specific cases can even make the code faster. He however mentioned that timing attack that exploits non-constant time integer arithmetic is not addressed by the proposed solution – e.g. for power PC processor. However, such attacks are not reported to happen in wild. He spent time to illustrate how load from and stores to addresses that depend on secret data, leak secret. A simple cache-timing attack does not reveal the secret address, it reveals just the cache line. He introduced constant-time equality comparison and used that to develop constant-time look-up table access from cache, which is effective for small tables. He mentioned that the effect of timing variance on oblivious RAM is worth investigating.

Audience showed tremendous interest in this topic. One question was on if we pre-load the table to cache, does it prevent the attack? Role of deliberate interrupt injection with potentially a fault attack in mounting timing attack was discussed and inferred as a topic which is worth looking further. Another question from the audience was: can you load part of the table (say half) to come to a good balance between performance and timing attack

resistance? The speaker agreed on that although pointed that compromise of security may not be acceptable in most applications.

Several other issues that were discussed are below. AES on composite field can potentially address the performance issue. Does Intel SGX architecture protect against timing attacks? They flush hashes between context switches. Yet it may not protect against timing attacks. Why do we see timing attacks in real world crypto algorithm every year? We see attacks against openSSL. Do we need better compiler? Do we need better verification? Is speed really that important? Small hit in performance should be tolerated. Do we rethink cryptographic algorithm? Symmetric crypto which does not leak timing information. Everyone agreed that we need better hardware support – e.g. basic integer operation is constant time to deal with the attacks. Can we come up with new instruction like cache locking that helps in preventing time channels? The security-performance trade-off needs to be considered in this context.

4.9 MAFIA: Micro-architecture Aware Fault Injection Attack

Bilgiday Yuce (Virginia Polytechnic Institute – Blacksburg, US)

Notes taken by Hirokata Yoshida.

Abstract: Fault attacks are a serious thread to secure embedded software running on a wide spectrum embedded devices. In a fault attack, an adversary breaches the security by injecting faults into the underlying processor hardware and observing their effects in the output of the running software. For fault injection, the adversary temporarily alters the execution of instructions by running the processor beyond its nominal operating conditions. Therefore, an efficient fault attack requires hardware-level and software-level knowledge of the target system.

In this work, we propose an instruction fault sensitivity model that systematically captures the fault sensitivity of the processor pipeline for different instructions. This model enables us to gain insight into the most likely faults during the execution of an instruction, and to pinpoint the most sensitive points during the execution of a program. We also introduce a fault attack methodology called Microarchitecture Aware Fault Injection Attack (MAFIA), which makes use of the proposed model. In MAFIA, the adversary analyzes the executing of the target software on the target processor to design and implement a fault attack. The adversary starts with an algorithm-level analysis to determine high-level attack objectives. Then, the adversary examine the target software at the instruction-level to determine potential instructions for fault injection. Finally, the adversary analyzes the cycle-accurate execution of the target instructions with the help of the fault sensitivity model to determine best clock cycles and fault injection parameters to attack.

We demonstrated the efficiency of the proposed method on a LEON3 processor implemented on a Xilinx Spartan6 FPGA. As the target software, we attacked instruction duplication countermeasure, in which the sensitive instructions of a program duplicated and the consistency of the results of both copies are checked. It is assumed that such a countermeasure can only be broken by injecting multiple identical faults with expensive fault injection tools. We broke this countermeasure with single clock glitch injections by creating an instruction fault model for the LEON3 processor and using MAFIA.

The key conclusion is that one needs to consider both hardware and software layers to design efficient countermeasure against fault attacks targeting the embedded software. Currently, we are working on hardware/software methods to protect our processors from this kind of threats. As the future work, we are also planning to investigate the efficiency of MAFIA on the hardware duplication based countermeasures as well as on the multicore systems.

Presentation Fault attacks are an important class of hardware oriented attacks. Basically, they inject the faults into the operation of the device. They analyze the response of the device of this fault injection to breach the security of the device. What is software fault attack? He first analyzes algorithm. He makes the assumption on the faults. We call this fault model. Let's take a close look at fault injection process. It is executed sequence of the instruction. The attacker changes the operation condition of the device e.g. the attacker can change voltage of the device, clock signal of the device. This faults injection affects execution of instructions then instructions are propagated to software. The current fault model does not consider the hardware aspects of microprocessor. This creates gap between injected faults and assumed faults.

We propose instruction fault sensitivity model for a RISC pipeline. Using this kind of method, we can pinpoint what kind of fault we will inject into the device. Our target countermeasure is instruction duplication countermeasure. Basically, in this countermeasure, to protect any instruction, we execute instruction twice and then we compare the results of these two executions of the same instruction. and if result doesn't match, we alarm wrong signal. Taking the example of 7 stage 32-bit pipeline, how attack works is shown. The effect of data dependencies is studied and it is shown that this leads to additional opportunities for fault injection. Instruction fault sensitivity model of each instruction is explained for a specific cycle. Memory stage of the load instruction takes 7.5 nano seconds. fetch stage takes 3.5 ns. So we make such a table for each instruction of the processor. After getting this model at once, for a processor, we can use this model several times to apply different patterns. After our experiments of work., we achieve all of our scenario and we break this instruction duplication countermeasure.

The conclusions are as follows: If we want to design efficient prototypes against embedded software, we need to consider both hardware and software aspects of this attack. Considering these aspects, we will see existing countermeasures are vulnerable against this new type of attacks. We need to countermeasure if we want to protect our devises against this kind of attack. Currently our recent work is trying to find a better way of protecting processor against this kind of attacks.

Discussions

1. What if instructions and hardware are duplicated? → It depends on the resulting architecture. It depends on the all the instructions in the pipeline.
2. What about other faults like EM faults. In this case, you need to change the fault sensitivity model. This case is timing.
3. Examples you showed just try instructions in pipeline, for the other things, you could also check some there. May be we find a good cycle to attack.
4. What if I use identical instructions? You are saying that to protect two more instructions, the requirement is to protect any one of them. Do you think this is a viable countermeasure?

4.10 Optical Interaction through Chip Backside with Nanoscale Potential

Christian Boit (TU Berlin, DE)

Notes taken by Shahin Tajik.

Although there is some basic interaction and attack possibility through the frontside of the chip, optical interaction with the active devices is not so fruitful, because the metallization layers obstruct the optical paths. However, the light which is reflected or emitted from the backside of the chip faces no optical obstruction. The utilization of the flip-chips is another motivation for us to access the chip through the backside.

Photon emission is detectable through the backside during switching events of transistors. With the help of this technique the IC can be debugged. Furthermore, the signal flow can be followed on the chip with the help photon emission and Picosecond Image Circuit Analysis (PICA). On the other hand, we need to stimulate some areas with lasers to interact and debug the chip. In this case, part of the light will be reflected back and part the light will be absorbed. The latter can creates voltage and current sources, which can leads to fault injection attacks and read-out of the data. Some techniques deploy wavelengths, which can just create heats but no electron-hole pairs in the silicon. This technique is called thermal laser stimulation (TLS). Using this technique one can read-out the stored values in the SRAMs. Note that there is no need for a clock signal. On the other hand, the reflected light will be modulated and can be used for the contactless probing of the signals. Different space charge layers on transistors are based on "on" or "off" state of the transistors and the reflected light has linear relations with voltage of the chip.

Photon interaction is bounded with the absorption of silicon for different wavelengths. Optical techniques have been taking advantage of the high infrared (IR) transmission for wavelengths $> 1\ \mu m$. One of the hardest challenges for optical IC debug techniques is the ever increasing miniaturization. 10 nm and smaller technologies are the current feature sizes. Resolution R in the optical interaction is a function of wavelength $\lambda : R\ \lambda/(2NA)$. NA is the Numerical Aperture (in air < 1) with $\lambda\ 1\mu m$. R is at best around 500nm if the chip is simply put into the optical path of the instrument. By Introducing solid immersion lens (SIL) on back surface, the NA is increased by the index of refraction nSIL. For silicon and $\lambda = 1\mu m$, n is about 3.5, resulting in a maximum R of around 150 nm. The smallest technology announced by Intel, which can be probed is 10 nm. Note that the technology length is the feature size of the transistor's gate and not the actual size of the transistor itself. The size of the FinFET technology is decreasing. In 2025 it will be 1.8 nm (pitch= 20nm). Slow decrease of the pitch helps us to still debug the chip.

If the resolution needs to increase further, the NA part of the equation cannot be enhanced more. Wavelength reduction remains the only possible approach. Below 10 nm we need to use shorter wavelengths. However, the absorption of silicon is problematic and most of the photons are absorbed by silicon substrate. We still can thin the substrate to 10 micrometer to reduce the absorption of the light. Moreover, in order to probe with visible light the silicon-based SIL should be replaced by GaP-based SILs.

4.11 How Secure are Modern FPGAs?

Shahin Tajik (TU Berlin, DE)

Notes taken by Bilgiday Yuce.

Most of the modern FPGAs keep their configuration in volatile, on-chip SRAM cells. Therefore, the configuration needs to be loaded into the on-chip SRAM cells from an off-chip Non-Volatile Memory (NVM) whenever the FPGA is powered on. In an untrusted environment, the transfer of configuration bitstream from off-chip NVM to on-chip SRAM cells may leak the design information. To mitigate this problem, FPGA vendors use bitstream encryption. In a trusted environment, the bitstream is encrypted and it is written into the NVM. Meanwhile, the encryption key is embedded into the FPGA. At each power-on of the FPGA in the untrusted environment, the encrypted bitstream is transferred to the FPGA from the NVM and it is decrypted using the embedded key. In this work, the target FPGA use soft PUFs for key storage and DPA-resistant decryptor for bitstream decryption. Although this approach provides protection against DPA and semi-invasive front-side attacks, it does not provide protection against semi-invasive back-side attacks such as Laser Voltage Probing (LVP) nad Laser Voltage Imaging (LVI). LVP enables us to probe electrical signals of interests by just pointing the laser beam to the circuit node of interest. LVI allows us to create a 2D map/image of the electrical nodes operating at a specific frequency, and filtering out the remaining electrical nodes operating on a different frequency. This work provides two LVP-based attacks against FPGAs during the configuration. In the first attack, an adversary probes the contents of PUF response and key registers with LVI. Therefore, the adversary can extract these values from the FPGA and decrypt the bitstream. The second attack uses a combination of LVI and LVP to characterize the PUF that is used as key storage. After characterizing the PUF, the adversary can clone its functionality and retrieve the decryption key. These two attacks were demonstrated on Altera Cyclone IV FPGAs. An existing method for protecting devices against laser-based attacks is using Silicon light sensors. However, the ligh sensors are ineffective against the LVP and LVI techniques proposed in this work because the laser beam has a larger wavelength than the silicon bandgap. A possible countermeasure would be assigning random values to registers in the case of a reset event. Using a special coating on the backside of the FPGA can be another alternative protection. This work also proposes use of a ring oscillator network as a countermeasure against LVP. In the proposed method, a network of ring oscillators with virtually equal frequencies are deployed on the FPGA. Using LVP will then shift the frequency of the ring oscillator in the vicinity of the probed area. Therefore, the frequency deviation from the average frequency of the ring oscillator network can be used to detct LVP and raise an alarm signal. Preliminary results for this detection technique seems promising and a research is currently going on. As a result, two backside attacks based on back-side LVP and LVI are proposed to reverse engineer the key of the FPGAs. This shows the need for countermeasures for this kind of attacks.

Questions/Comments:

- Q: Why do PUFs are a better choice for key storage than BBRAMs or eFuses?
- Q: How to make sure PUF is really loaded correctly? Is it possible to change bitstream?
- Q: Would aging of PUF be a problem from the key storage point of view?
- Q: If you deploy ring-oscillator-based detector through the whole chip, they would cause a significant heating. Would it effect the measurements?
- Q: How big is the laser machinery used for LVI and LVP?

- Q: How many measurements are need to create LVI map of the electrical nodes of interest?
- Q: How quickly can you turn on/off the ring-oscillator-based detector?
- Q: How many ring oscillators did you use in ring-oscillator-based detector?

4.12 Detection and Prevention of Side-Channel Attacks

Naofumi Homma (Tohoku University, JP)

Notes taken by Debdeep Mukhopadhyay.

- Local EM attacks
  - Using Microprobes, observation of precise and local EM leakage
  - Beyond conventional leak assumptions
- Measurable leaks by microprobes
- Most of the countermeasures can be defeated because their leak assumptions are not met
- Countermeasures: Transistor level balancing, active shielding, special packaging
- Overhead, vulnerability still exists, use high resolution, or reverse-side attacks
- EM attack sensor-CHES 2014
- Idea: sense the presence of probing by observing electrical coupling, EM field variation LC oscillation frequency shifts due to Mutual Inductance, M.
- Based on this idea, proposal of Dual-coil sensor architecture
- No frequency reference needed. Oberve variance of the coil oscillation frequency, f_LC
- Sensor core architecture uses two coils. The detection circuit subtracts the two coil oscillation frequencies.
- The experimental set up was elaborated fabricated in 0.18 u CMOS.
- Detection, probe diameter 0.2 mm, 0.3 mm.
- Greater than 1% variation in f_LC can be detected. The detection range is 1 mm.
- Overhead: AES core 24.3 K, sensor 0.3 K, overhead is 1.2% Power: +9%, lesser than classical countermeasures

Limitations

- Attack may be to keep difference of LC oscillation frequencies during measurement, but attacker cannot see oscillation frequency.
- Detection vertical distance is 0.1 mm. Should be ok for front end attack. Conventional EMAs over chip package still possible. Combination with classical countermeasures important.
- Scaling on EM attack sensor: Other non-crypto algorithms: could compensate algorithm/gate-level countermeasures. Also applicable for other platforms, FPGAs, and even advanced CMOS technologies.
- Advanced CMOS: Oscillation frequency would increase. Magnetic flux passing through probe would decrease is the probe diameter is smaller.
- Consequence of smaller probes: Frequency shift amount would decrease: Digital counter may not detect the frequency shift.
- Improvement: Extend detection time. Extend detection process time to accumulate smaller shift amount differences.

- Time extension enables to detect small frequency shifts even probing from back side of LSI, arund 0.5 mm.
- Overhead: Additional bits are required to counter and subtractor in addition to time delay, Delay: +12.1%, Area: +1.6%.
- Cancel out timing overhead by Simultaneous Operations of Crypto Core and Sensor
- Works because current flowing in crypto core minute and omnidirectional.
- Another idea for speed up: Frequency count by Time to Digital Converted (TDC)
- Conclusions: Sensing technology for side channel attacks
- Challenge for scaling: Application to other platforms, other technologies.

Questions

- Which Microprobes: 0.1 mm diameter hand made microprobes., Langer probes.
- Can be used for laser fault detection?
- Detection coils of the size does it depend on the probe size?
- EM attack sensor
- Scaling on attack sensor: What may happen in advanced CMOS technology

4.13 Implementation Security through Dynamic Reconfiguration

Nele Mentens (KU Leuven, BE)

Notes taken by Patrick Schaumont.

Research Interests

- Efficient Crypto Coprocessor Design
- Design automation/ design space explo for crypto hardware
- Partial reconfiguration for security purposes

Implementation Security through dynamic reconfiguration

Why?

- Power Analysis attacks correlate power and secret data Fault Analysis attacks correlate faults and secret data
- Approach is to make the hardware dynamically reconfigurable without changing the IO behavior of the system

Randomly Reconfigurable Architectures

Symmetric Key Algorithms

- Randomized Pipelining within a round
- Randomize Pipelining within an SBOX

Move pipeline registers around to randomize power dissipation

Q(Patrick): Can we accumulate power over sufficient clock cycles to remove randomization effect?

Public Key Algorithms: Many possible randomization elements
- Parameters
- Circuit
- Order of Operations, randomized addition chains

Randomized ECC 25519 Q(Peter): For Montgomery representation? Yes

The design space for PK Algorithms is much larger, so you can use evolutionary algorithms and design automation to search feasible solutions

E.g. Randomized Addition Chains can be synthesized

Reconfigurable Technology. SRAM Configurarion Memory: Partial FPGA reconfiguration, such as in Xilinx, allows to reconfigure part of the FPGA. The new FPGA-SoC allows the FPGA to be reconfigured from within the processor. SRAM reconfiguration is coarse grain, with minimum width and height in the FPGA fabric. This is relatively slow because of serial (sequential) loading of partial bitstream.

CFGLUTs: 5-1 LUTs. Can be reconfigured directly by the user logic. This is very fast: 32 cycles to reconfigure one LUT. Routing cannot be reconfigured this way.

Virtual Reconfigurable Circuits: Circuits specifically designed for dynamically reconfigurable designs (Coarse grain reconfigurable asics)

Generation of New Configurations
- Offine generation: a fixed number of configurations stored in SRAM
- Online generation: (seems) not feasible at this moment for realistic designs. However, dedicated designs using CFGLUT may be feasible.
- Parametrizable bitstreams

Summary of Design Parameters
- Reconfiguration Time
- Reconfiguration Overhead
- Reconfiguration Granularity
- Reconfiguration Frequency
- Target Platform
- Number of Configuration Options

Q(Patrick): Connection to Whitebox Crypto? A: Whitebox does not work (or is obscure)

Q(Tahin): Can the attacker tamper with reconfiguration? Partial bitstream is well defined (in position), so could be tampered in principle.

Q(Roy): Can this be used for Trojans?

4.14 Propagation of Glitches and Side-channel Attacks

Guido Bertoni (ST Microelectronics – Agrate, IT)

Notes taken by Ingrid Verbauwhede.

Glitches represent a great danger for hardware implementations of cryptographic schemes. Their intrinsic random nature makes them difficult to tackle and their occurrence threatens side-channel protections. Although countermeasures aiming at structurally solving the problem already exist, they usually require some effort to be applied or introduce non-negligible

overhead in the design. Our work addresses the gap between such countermeasures and the naïve implementation of schemes being vulnerable in the presence of glitches. Our contribution is twofold: (1) we expand the mathematical framework proposed by Brzozowski and Ésik (FMSD 2003) by meaningfully adding the notion of information leakage, (2) thanks to which we define a formal methodology for the analysis of vulnerabilities in combinatorial circuits when glitches are taken into account.

4.15 Treshold Implementations

Svetla Nikova (KU Leuven, BE)

Notes taken by Chen-Mou Cheng.

Masking is not secure in CMOS because of glitches. TI: provably secure masking scheme based on secret sharing and multiparty computation. It was initially proposed for 1st order DPA but later on extended to any order. It is secure even in face of circuit glitches.

TI conditions: Correctness, non-completeness, and uniformity. TI techniques: It is important to decompose a complicated (high-degree) function to reduce its degree (and hence circuit complexity). There are automatic tools for decomposing and sharing. Other important optimizations include reusing and factorization.

AES: success; Keccak: efficient implementation proposed, ongoing work to solve uniformity issues. TI has also been applied to other ciphers such as Katan, Simon, Speck, ...

Higher order TI: any combination up to d component functions must be independent of at least one input share. Naturally, the number of shares increases. The cost for higher order TI is roughly linear in the degree for Katan-32 but can increase more rapidly for other, more complicated ciphers such as AES. Often time we can trade off between area and the amount of required randomness.

Discussions

Q: Is it possible to combine RNG & TI to make it more efficient?
A: It is possible to reuse randomness, e.g., in TI of Keccak. However, need to be careful about entropy & dependencies, etc.
C: Would be interesting to compare against glitch-free circuits in terms of area and randomness costs.
C: In practice, crosstalk can decrease the security of TI (and also other masking schemes).

4.16 Secure Scaling, Scaling Securely

Francesco Regazzoni (University of Lugano, CH)

Notes taken by Patrick Schaumont.

Secure Scaling, Scaling Securely

Handling the Scaling

- We need Design Automation. Early chips were designed by hand. Modern chips are designed with extensive use of design automation.
- We need Design Auotmation for Security. Security is considered at the end of the design. Cost and Time to Market are most important. Avoid Security Pitfalls. Handle the Complexity. And, most importantly, use standard design commodities (= tools).
- Automatic Application of Countermeasures. Input = Unprotected Algorithm + Countermeasure. Output = Algorithm where the countermeasure is applied. That does not mean that the result is a protected algorithm. It is only protected if the countermeasure is correct.
- Example: Software Automation with Compiler. Start from Software Implementation, do Information Leakage Analysis, Transformation Target Identification, do Code Transformation.
- Example: SC Leakage analysis by mutual information analysis. Then transform the instructions.
- Example 2: Protection using Custom Instructions. Implement Custom Instructions using protected logic.
- Example 3: Verification. Input: Algorithm, countermeasure. Output: Check that countermeasure is correctly applied.
- E.g. masking of an expression

$$s = p \operatorname{exor} (k \operatorname{exor} m)$$

$\rightarrow$ compiler produces

$$s = (p \operatorname{exor} k) \operatorname{exor} m$$

which is wrong

- Example describes SLEUTH.
- Risks of Scaling
- Can fault attacks be applied to subtreshold technology?
- Subthreshold is used for low power.
- Fault attack: 0.8mV interval enables single-byte fault
- The interval depends on chip and temperature. In subthreshold, higher temperature gives lower threshold voltage.
- Scaling Securely
- Current photon based entropy source
- Initial Tests
- 512 x 128 single photon detectors Photon passes through semitransparant mirror.
- Parallel readout and Von Neumann Postprocessing
- Initial Tests pass the NIST tests when Von Neumann is included.

4.17 Scaling of Implementation Attacks

Georg Sigl (TU München, DE)

Notes taken by Francesco Regazzoni.

Advanced EMA attacks Advanced Laser Attack Countermeasures

- Attack Setup: The station for mounting attacks has a fixed probe and a moving base to position the FPGA. The attack is carried out on the FPGA from the front.
- Advanced EMA attacks: The target is an assignment operation during an ECC computation. The values are stored in registers, which have a physical position on the die. If the probe is placed very close to the register storing value A, there will be a significant variation on the probe when there is a change in the value of A. If there is a dependency of the register usage and the secret information, this can used far an attack.
- In the case you will have N iterations depending on the length of the secret, it is possible to split the traces collected into N portions, each of them correspond to a calculation of a single bit. The traces are then assigned to two different sets to recover the secret key.
- The starting point of the attack is the identification of the hot spot for measurement (the points which allows to get the key). This cartography operation requires some time.
- Principal Component Analysis can improve the key recovery process. The PCA generates N principal components (where N is the length of the secret). If components clearly divide the traces into two clouds then the key is recovered easily. Reported results show that component 4 produces the best result.
- Single vs multiple probe (3 probes): with multiple probes, more PCA components contain information.
- With scaling of technology, EM side channel will be still feasible. If scaling means using better and more probes, EM is likely to be more dangerous.
- Advanced laser attack: A two laser setup has been internally developed, emitting beam for attacking front side and another beam for attaching back side. The attack was carried out on SPARTAN-3 at 90nm and SPARTAN-6 at 45nm. It is still possible identify block RAMs and flip single bit at 45nm, but, comparing with the 90nm it is getting harder.
- An attack on AES with the infection countermeasure has been performed. To attack this countermeasure it is needed to shoot the laser at both computation units, or at the comparator. The target was SPARTAN-6. To identify the position where to shoot with the laser it is needed to generate a map with block RAM and flip-flop (which is design independent) and the flip flops used in the applications (which is design dependent). From over 80'000 shoots, 229 were exploitable faults (remember that even a single exploitable fault is sufficient).
- Attack summary: attacks are feasible also for smaller technologies, although shrink would make the shoot less precise. Also laser can scale by increasing the amount of lasers, e.g..
- As countermeasures two approaches can be applied: Attack detection (sensors, detectors, error detectors) and Attack prevention (Limiting the amount of repetition, add randomness, hiding). Both have to be taken into account vertically, spanning over cryptographic algorithms, System, Technology.
- Each countermeasure should address the following questions:
  - What is the coverage?
  - Which one is the best layer to implement it?
  - Is it possible to implement an orthogonal countermeasure?

4.18 Crypto, Integration, Technology: Good, Bad, Ugly?

Debdeep Mukhopadhyay (Indian Institute of Technology – Kharagpur, IN

Notes taken by Sahin Tajik.

Talk Abstract: The talk presents a glimpse on the effects on physical security as a result of integration, and technology scaling. Technology scaling and improvements in computer architecture have varying effects on side channels: 1) Good, when the attacks are hindered, 2) Bad, when the attacks are aided, and 3) Ugly, when it is difficult to characterise the consequences! The talk presents few case studies to illustrate the dependencies. Differential Fault Intensity Attacks (DFIA) is a menacing class of fault attack against cryptosystems. Fault attacks are typically countered using concepts of redundancies, which modern computer architectures seem to support owing their increases parallelism. However, malicious fault attackers with the capability of repeating fault injections with precise control can potentially defeat such classical countermeasures. In this context, Fault space transformation (FST) is proposed as a new class of countermeasures against these evolved fault attacks. The objective of FST is to reduce the probability of obtaining useful faults which can bypass standard countermeasures. Although, device scaling increases the failure rates, the chances of obtaining useful faults wrt. FST are further reduced, to make FST a promising fault attack countermeasure. The talk also shows an example of cache timing attack on a 128 bit cipher, known as Clefia. The cipher has small tables which reduces the chances of a cache attack, as the number of cache misses are a constant per encryption. However, modern computer architectures provide artefacts for parallel service of cache misses. The structure of block ciphers provide opportunity for out of order loading of tables in a round, but not across the rounds due to dependencies. This leads to timing variations because though the total number of cache misses are constant, the penalty due to cache misses in a round can be ameliorated compared to cache misses across rounds. Thus the distribution of cache misses also plays a role to determine the information leakage, and it was shown that modern computer architectures (after Intel Pentium-3) were prone to cache timing attacks on Clefia. Finally, the talk comments on the recently discovered bug on DRAM chips, typically observed post 42 nm technology. Repeated discharging and recharging of the cells of a row in a DRAM bank results in leakage of charge in adjacent rows. If repeated enough times, typically before the automatic refresh in adjacent rows, causes flips of bits. This example typically shows an instance where process integration leads to new avenues or sources of attacks. In conclusion, the talk tries to encourage the study of these effects to evolve systems which are more secured against the powerful side channel attacks by taking advantage of modern day architectures, while being aware of the vulnerabilities introduced by them.

- We would like to evaluate the physical security of crypto across integration and technology. We consider two cases: 1. cache attacks ,2. fault injection attacks.
- Cache memory leaks information based on a cache hit. In this case, the access time and power consumption will be less than the case, where there is a cache miss. As a result, the attacker can launch a cache attack by measuring the total time for the encryption. This technique has been used to attack a remote server. For instance, in the Bernstein's cache timing attack, we try to invoke the AES encryption by .xing part of the input, and randomize other parts of the inputs and obtain the total time for the encryption. By guessing the key and calculating the time correlations,we can break the AES. Smaller table

sizes make the cache attack harder. However, it has been shown that the cache attacks are possible even on the ciphers with small Sboxes. It has been shown that the CLEFIA cipher can be attacked. The processor's aim is to reduce the true miss penalty by using speculative loading, prefetching, out-of-order loading, parallelization and overlapping. However, the attack has been tested successfully on some platforms, such as Intel Core 2, Intel Atom, Pentium 4, Xeon – core 2 servers. However, the attack against Pentium 3 was unsuccessful.

- On the other hand, cryptographic algorithms can be analyzed using faults. Concurrent error detection, infection and data encoding can be the countermeasures against faults. However, naive redundancy can be broken by improving fault collision probability. A smaller fault space enhances the fault collision probability. A non-uniform probability distribution of the faults in the fault space also enhances the fault collision probability. With increase in the bias, the collision probability increases. Transforming the fault space implies that the adversary cannot beat the countermeasure by merely introducing the same fault twice. It is most unlikely that the transformed fault space will have a one-to-one correspondence in terms of the bias with the original one. Mathematically, the expected fault collision probability over all possible transformations is the same as for the uniform fault models.
- But what is the impact of technology scaling on the failure rates? Failure rate of a 65 nm device is 316% times more than the same at 180 nm. However, the growing parallelism can give rise to several levels of redundancy. Natural hardware faults can be detected by dual-modular-redundancy (DMR) and triple-modular-redundancy (TMR). Moreover, malicious faults can be detected by Fault Space Transformation. Fault Space Transformation can be applied on different cores exploiting the available redundancies.
- RowHammering: Repeated discharging and recharging of the cells of a row results in the leakage of charge in the adjacent rows. If it is repeated enough, typically before occurrence of the automatic refreshment in adjacent rows, causes .ips of bits, which is known as RowHammer. This development appears to coincide with the upgrade to 42 nm productions for the DRAM chips. A small cell can hold a limited amount of charge, which makes it more susceptible to data loss. The close proximity of the cells introduces electromagnetic coupling effects. Higher variation in the process technology increases the number of the outlier cells that are susceptible to the cross talks.This is an example of converting a reliability issue to an attack.
- We conclude that the cryptographic techniques often need to be considered based on the underlying platforms: Improved architectures may give bene.ts or open new threats. We propose Fault Space Transformation as a novel fault tolerance technique for the block ciphers. Parallel architecture offers opportunities for redundancy based schemes. The utilization of Fault Space Transformation may be a good idea! Finally, technology scaling offers more variability and improved failure rates: The controlled usage of the faults is a major challenge and it could lead to suitable hammers to create faults and threaten actual systems.

4.19 Smart Card Secure Channel Protocol

Joan Daemen (ST Microelectronics – Diegem, BE)

Notes taken by Patrick Schaumont.

Joan described a protocol that is used to implement the 'secure channel' of a smart card. A secure channel is the control/data link between the smart card and a central smart card management system (bank, network provider in case of SIM, ...). The entity on the card that implements the card end of the protocol is called "security domain". At the central system there is a hardware security module (HSM) that performs the central server side of the protocol.

The objective of the secure channel protocol is authorize commands send to the card, from the HSM to the security domain, and to authenticate responses returned from it, from the security domain to the HSM. Furthermore, the secure channel protocol enables the secure transfer of keys (e.g. installing a new vendor key). Some background information may be found from the GlobalPlatform Wiki (https://sourceforge.net/p/globalplatform/wiki/Home/)

The cryptographic primitive for authorization and authentication is a message authentication code (MAC). For authorization, the MACs are generated by the HSM and verified by the security domain; these MACs are called C-MAC. For response authentication, the MACs are generated by the security domain and verified by the HSM; these MACs are called R-MAC.

- The basic protocol for authorization (C-MAC) is a chain of MAC, where a C-MAC is repeatedly computed over new input data, which can be either a command or a response, and a secret MAC key. Hence, the complete chain of commands, or responses, is chained. Any tampering with the chain can be detected by the security domain. The computing of a C-MAC chain also enables to integrate non-functional context data, such as additional application identifiers and nonces.
- The basic mechanism for authentication is a single R-MAC, computed over the sequence of command-responses generated by the security domain. The R-MAC is verified by the HSM.

The protocol has several particular features compared to text-book challenge/response protocols.

- The protocol makes use of predictable nonce-counters, embedded in the MAC chain, which prevent replay of commands or responses. However, the protocol does not make use of random numbers and does not have freshness. Every C-MAC sequence is fully predictable. On the other hand, the R-MAC sequence contains freshness, since the command data (delivered through C-MAC) can include a challenge.
- The card has a key hierarchy which generates a sequence of 'working keys' for RMAC as well as for CMAC. These keys are derived from a master key using a one-way function, the card-id, and the C-MAC sequence counter. This means that the working keys are predictable, but also continuously updated. The working keys are stored in non-volatile memory and derived at runtime when needed. A new R-MAC working key can only be created when the C-MAC sequence was successfully completed.
- There is a ratification on the working keys, both R-MAC and C-MAC. The working keys are blocked when there is an excessive number of MAC failures (either R-MAC or C-MAC). Once the working key is blocked, the secure channel can no longer be used.

Participants

- Debapriya Basu Roy
Indian Institute of Technology – Kharagpur, IN
- Lejla Batina
Radboud Univ. Nijmegen, NL
- Guido Bertoni
ST Microelectronics – Agrate, IT
- Swarup Bhunia
University of Florida – Gainesville, US
- Christian Boit
TU Berlin, DE
- Chen-Mou Cheng
National Taiwan University – Taipei, TW
- Joan Daemen
ST Microelectronics – Diegem, BE
- Jia Di
University of Arkansas – Fayetteville, US
- Thomas Eisenbarth
Worcester Polytechnic Inst., US
- Naofumi Homma
Tohoku University, JP
- Yier Jin
University of Central Florida – Orlando, US
- Nele Mentens
KU Leuven, BE
- Debdeep Mukhopadhyay
Indian Institute of Technology – Kharagpur, IN
- Ventzislav Nikov
NXP Semiconductors – Leuven, BE
- Svetla Petkova-Nikova
KU Leuven, BE
- Bart Preneel
KU Leuven, BE
- Sandip Ray
NXP Semiconductors – Austin, US
- Francesco Regazzoni
University of Lugano, CH
- Kazuo Sakiyama
The University of Electro-Communications, JP
- Patrick Schaumont
Virginia Polytechnic Institute – Blacksburg, US
- Peter Schwabe
Radboud Univ. Nijmegen, NL
- Georg Sigl
TU München, DE
- Shahin Tajik
TU Berlin, DE
- Ingrid Verbauwhede
KU Leuven, BE
- Hirotaka Yoshida
AIST – Tsukuba, JP
- Bilgiday Yuce
Virginia Polytechnic Institute – Blacksburg, US