Power Consumption Profile Analysis for Security Attack Simulation in Smart Cards at High Abstraction Level[*]

K. Rothbart, U. Neffe, Ch. Steger, R. Weiss
Institute for Technical Informatics
Graz University of Technology
Inffeldgassse 16/1, A-8010 Graz, AUSTRIA
+43 316 873 6412
{rothbart, neffe, steger, weiss}@iti.tugraz.at

E. Rieger, A. Muehlberger
Philips Austria GmbH. Styria
Mikron-Weg 1
A-8101 Gratkorn, Austria
+43 3124 299 541
edgar.rieger@philips.com

ABSTRACT

Smart cards are embedded systems which are used in an increasing number of secure applications. As they store and deal with confidential and secret data many attacks are performed on these cards to reveal this private information. Consequently, the security demands on smart cards are very high. It is mandatory to evaluate the security of the design but this is performed often very late in the design process or when the chip has already been manufactured. This paper presents a hierarchical security attack simulation flow for smart card designs where security attacks can be simulated in the processor specific model at transaction layer 1 in SystemC. Therefore, the power consumption profile is analyzed at this level. Preliminary results show that this analysis at high abstraction level can be used to determine vulnerabilities of the system to security attacks. Moreover, points to insert software countermeasures can easily be identified.

Categories and Subject Descriptors

C.3 [**Computer Systems Organization**]: Special-purpose and Application-based Systems – *Smartcards.*

General Terms

Security, Design.

Keywords

embedded security, smart card, attack, fault injection, power profile, SystemC, simulation, analysis.

1. INTRODUCTION

Smart cards are very small computing platforms and used in an increasing number of mobile applications. Most of these applications such as SIM cards of GSM mobile phones, e-purses of bank cards and contact-less identification cards have very high security demands. The confidential data such as secret keys, personal data or private application information stored on these cards have to be protected against unauthorized access. As hackers want to get knowledge of this secret data they perform different security attacks on smart cards [1]. These attacks address the vulnerability of system components to reach the hacker's goal. Security attacks on smart cards can mainly be divided into fault induction and side-channel attacks. It is very important to test the robustness of smart cards against security attacks already in the design process and not when the chip has already been manufactured. This can save time and money as design changes can be made much easier at higher abstraction levels of the design. Moreover, it is important to support the design flow with security evaluation facilities at different levels of abstraction to be able to add countermeasures against security attacks at the current level of the design process.

In this paper a novel methodology for hierarchical security attack simulation at different levels of abstraction is presented. This includes the fault injection at functional level (FL) and transaction layer 1 (TL1) in SystemC [2]. The focus is on the integration of an instruction set simulator (ISS) with power consumption estimation capabilities to facilitate the analysis of the impact of faults on the system behavior. Moreover, the vulnerability of the system to certain power analysis attacks can be examined. This early analysis of the system security can help to identify points to insert software countermeasures to increase the robustness of the power profile against attacks.

The remainder of this paper is organized as follows. Section 2 surveys related work. In Section 3 the importance of extensions of the smart card design flow is described. Section 4 describes the hierarchical security attack simulation methodology. Experimental results are presented in Section 5. The last Section is devoted to concluding remarks and further work.

2. RELATED WORK

In [3] the authors concentrate on smart card design with complementary hardware and software which eases the development of highly secure systems. They pointed out the importance of the role of smart cards as secure storage and the capabilities for cryptographic computations but also the needed flexibility of the system. Furthermore, they described hardware and software countermeasures against security attacks but do not take into account any testing for security before silicon neither did they proposed any design flow.

New methodologies in smart card security design are presented in

EMSOFT'05, September 19–22, 2005, Jersey City, New Jersey, USA.
Copyright 2005 ACM 1-59593-091-4/05/0009...$5.00.

[*] This work was funded by the Austrian Federal Ministry for Transport, Innovation, and Technology under the FIT-IT contract FFG 806014.

[4]. The author proposed a methodology based on concurrent secure development using a top-down design. The objectives of this approach are to decrease development time and to integrate and assure the security requirements for evaluation all over the development cycle. They used the most advanced techniques of semi-formal and formal methods like UML and B for security checking. Furthermore, they defined translation methods from the informal security requirements to a formal model. In our approach simulation of security attacks is performed to enable security evaluation already at functional level.

Fournier et al. presented in [5] a security evaluation of an asynchronous smart card system. They applied different side-channel attacks and performed fault injection analysis on a smart card chip. They also outlined a research program to investigate design time security validation techniques and described the Design-time Analysis. In this approach they emphasize the importance of simulating the system behavior during the design process in order to evaluate the security. They used a gate level power estimation tool to simulate side-channel information leakage and fault attacks.

In our approach security attacks on smart cards can be simulated using fault injection at three high levels of abstraction and therefore enables testing for security already at design-time. Security analysis at design-time rather than relying on post manufacture analysis can decrease the time for design as silicon re-spins can be prevented.

3. SMART CARD DESIGN FLOW

As outlined before the importance of simulation for security is obvious. Usually countermeasures against security attacks are considered already at higher levels of abstraction of the smart card design [6]. But as smart card systems are getting more and more complex simulation and testing of the system robustness is also necessary. Moreover, designers also often rely on hardware countermeasures like tamper sensors for power or frequency glitches, temperature and light. But as described in [1] several of those hardware countermeasures can be overcome by more sophisticated attack techniques. Furthermore, software countermeasures can assist existing hardware countermeasures to protect the system against future attack techniques. When security is only tested on the manufactured chip the insertion of additional countermeasures is very likely and as a consequence redesigns are needed. This means that designers have to go back to high abstraction levels, insert the countermeasures, and run the whole design process again to later examine the effectiveness of this action. Moreover, some tests for security are quite difficult and time-consuming for post manufacture analysis and ergo costly.

4. HIERARCHICAL SECURITY ATTACK SIMULATION

To overcome the problems mentioned before security should be analyzed already at design-time. Especially in the case of smart cards means to examine the vulnerability against security attacks at high abstraction level of the design process are needed. Furthermore, the higher the level of abstraction is where security evaluation takes place, the easier and faster countermeasures are set and their effectiveness evaluated. Many security attacks on smart cards lead to faults in the system. Hence, all those attacks can be simulated at high abstraction level using fault injection. As depicted in Figure 1 the earliest stage of the design process at which attacks can be simulated is the functional level. According the design flow interface generation and system synthesis is performed. The transaction layer 1 (TL1) is divided into two stages to provide better design-space exploration with the intermediate platform model (IPM). The IPM is achieved by mapping a functional model onto an architecture model. The second stage of TL1 is the processor specific model (PSM).

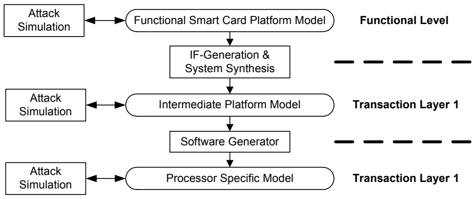

Figure 1. Overview of the hierarchical security attack simulation.

Attack simulation and therefore security evaluation is possible at all three levels. This is important as different details are available at each stage. Hence, different security attacks can be simulated at the highest possible level of abstraction.

4.1 Attack Simulation at Functional Level

The functional model consists of functional blocks which are connected to each other over ports and interfaces. Fault induction attacks can be simulated at this level using fault injection in SystemC. Fault injection modules and fault injection ports are used to inject faults at a specified location into the system [7]. To analyze the system behavior the design has to be instrumented. By the insertion of pre-processor macros the behavior can be traced. The smart card description in SystemC is automatically instrumented using a predefined security level which controls the location and amount of the instrumentation. Then, the system is prepared for fault injection [7] and both the faulty and the regular system are simulated. Next, evaluation of the analysis data takes place.

4.2 Attack Simulation at the Intermediate Platform Model Level

At the TL1 bus communication is cycle accurate but the simulation time is still much shorter than at Register Transfer Level (RTL). At this level a bus control unit is used for bus communication. Faults can be injected again using special fault injection ports. Security attacks on the bus can be simulated at this design stage [8]. The same methodology for behavior analysis as at functional level is used. Thus, again the system behavior can be traced applying instrumentation.

4.3 Attack Simulation at the Processor Specific Model Level

The platform used is a smart card based on a RISC core from MIPS Technologies, Inc [9]. It is not feasible to insert statements for instrumentation purposes into the processor specific model to trace the system behavior as it is performed at the higher abstraction levels. This would change the system behavior. But it is important to trace the system behavior in the presence of faults in order to analyze the robustness of the system against security attacks also at this abstraction level. To overcome this problem an ISS [10] for the high-performance smart card CPU [9] is used. This ISS executes cross-compiled code. Therefore, the functional modules mapped on microprocessor units have to be converted into target specific C code. The ISS decomposes the energy consumption into an instruction and data dependent part and outputs the energy consumed per cycle. This output is used to trace the system behavior. The characterization for the design was based on the smart card architecture mentioned before. A prototype and the entire database were available to evaluate the estimated power consumption. Therefore, very accurate energy estimations are feasible. It is important to rely on the power information as the power profile is used at this abstraction level to evaluate the system robustness against security attacks.

For the security evaluation using the power consumption profile different ways of their comparison have to be considered.

4.3.1 Simple Comparison

First, a simple comparison of the power values of every cycle is performed to analyze the effectiveness of faults. Thus, the security attack simulation methodology at the higher abstraction levels can be evaluated. Moreover, again the system behavior can be analyzed as it is done at the higher levels. Figure 2 shows the fault injection process into the platform specific model at TL1. The fault injection script file includes information like the fault kind, the occurrence of the fault in matters of location and clock cycle. The fault injection controller injects faults into the model according to the fault injection script file by using fault injection ports and the system clock. A power data file is generated while the smart card specific model is simulating. This data file contains the power value of every clock cycle, the program counter information, and the number of the injected fault. The value of the program counter helps to localize the resulting error in the program running on the instruction set simulator. This eases the insertion of countermeasures as the exact location of injected fault and the occurred error is known. Often only the system or module outputs are compared to the correct data which makes it sometimes more difficult to find the exact location of the error and also the clock cycle when the error occurred. The graphical output of the power profile can be used as well to see the impact of certain faults rather than only power data. To analyze the correctness of the system behavior after faults have been injected all power data files resulting from the simulation runs with injected faults are compared to a golden power data file. This golden power data file is generated by a simulation run without the injection of faults assuming that the system is functionally correct. Then, the power profile comparator compares the power data files.

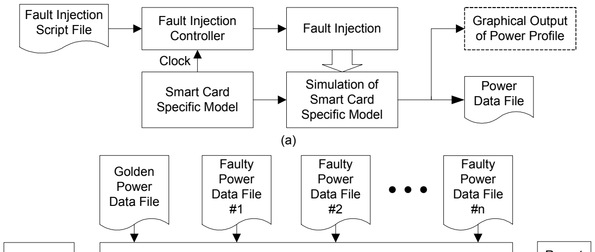
Fault Injection Script File
Fault Injection Controller
Fault Injection
Graphical Output of Power Profile
Clock
Smart Card Specific Model
Simulation of Smart Card Specific Model
Power Data File
(a)
Golden Power Data File
Faulty Power Data File #1
Faulty Power Data File #2
Faulty Power Data File #n
Sensitivity
Power Profile Comparator
Report
(b)

Figure 2. Fault injection into platform specific model (a) and analysis of the system behavior using the power profile (b).

4.3.2 Comparison of the Filtered Profile

Second, a pre-calculation of the power values prior to the comparison is performed. Therefore, a low-pass filter is applied as results of measurements show this behavior of the prototype. As illustrated in Figure 2 (b) the power profiles are compared according to a sensitivity level. This sensitivity level is used to define how big the deviation of the correct profile can be without writing it into the report file. Moreover, minor errors like data value errors which do not result in control flow errors can be neglected with this. After comparison the power profile comparator writes the discrepancies into a report file recording the program counter value, power value, clock cycle and the fault number. Figure 3 illustrates an example of the comparison of a correct power profile to a faulty one. With the sensitivity the area of the tolerated power values can be defined. If a fault is injected which results in an error causing a power profile that exits this area, the error will be detected by the system. This analysis method can be used to detect vulnerabilities of system components to security attacks which induce faults.

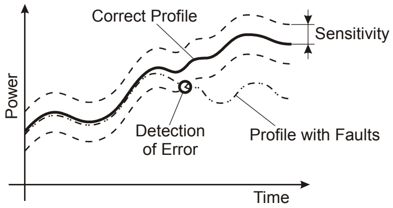

Figure 3. Example of the comparison of a correct power profile to a faulty one.

Furthermore, the automatic analysis exposes divergences in the power consumption profiles of different implementations. This is important to set fast and easy countermeasures related to the power profile like instruction reordering or the insertion of dummy instructions.

4.3.3 Correlation of Profiles

Another important topic is the correlation of power profiles. Simple comparison of power consumption profiles might often not be enough to gain sufficient information. In case of, for example, cache misses the resulting power profile changes in a way that simple comparison would show total discrepancies. Therefore, it is important to calculate the correlation of the profiles to find synchronization points. This reveals if a system can recover from an occurred error and continues its correct operation. Thus, the normalized cross-correlation is calculated when the compared power profiles have differences over a long time in contrast to the profile lengths.

4.3.4 Analysis in the Frequency Domain

As sometimes the frequency domain reveals more relevant information than the time domain a transformation between the two domains can be performed as well with the environment presented in this paper.

5. EXPERIMENTAL RESULTS

For first results evaluating the methodology for the behavior analysis at transaction level 1 of the processor specific model a PIN verification algorithm has been chosen. Figure 4 shows an excerpt of the power profile of the PIN verification algorithm. A security attack like on the data bus during information transfer has the effect of reading the value 255 (0xFF) regardless of the transferred information's actual value. This kind of fault, resulting in reading all the bits of a variable as "1", has been injected. That leads to higher power consumption in one clock cycle which is pointed out in the figure. This shows that the power consumption is highly data dependent and therefore, needs to be taken into account. Moreover, this should illustrate that such a single data fault will be masked by the low-pass behavior of the chip and might not have further effects.

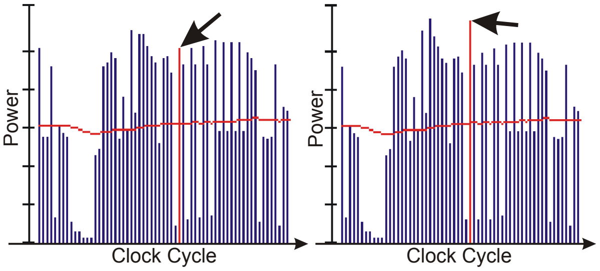

Figure 4. Excerpt of the power profile of PIN verification. (a) correct profile, (b) profile due to fault injection with different data.

For further evaluation purposes another application has been used. This program implements an error handling routine which is used in a similar way in the Java Card^{TM} Virtual Machine (JCVM) [11]. As it is illustrated in Figure 5 the normal execution of the application (regular profile) consumes roughly a constant power. This shows that the implementation of this application is quite robust against power analysis attacks as certain attacks aim to determine when, for example, a branch instruction is about to taken.

Then, a security attack has been simulated by setting one bit in the memory to stuck-at zero. This results in a different execution path to be taken which is executed in a loop without throwing an exception. Analyzing this power profile one can easily determine when a subroutine call is performed. These high differences between the peaks result from the high initialization effort of the subroutine and the execution of operations which consume less power than in the regular execution. With the information obtained from the power profile an attacker can refine his attack to target directly the subroutine call to force jumps to other memory areas in order to execute malicious code.

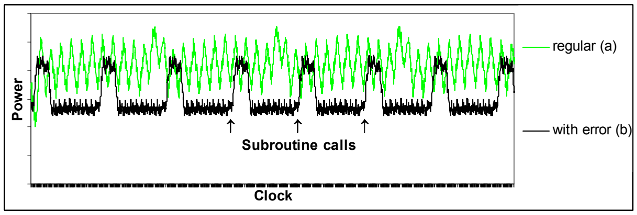

Figure 5. Excerpt of the power profile of an error handling routine. (a) regular profile, (b) profile with error.

When simulating security attacks at such a high level software countermeasures can easily be set to blur the power profile and harden the system. This can be achieved by the insertion of dummy instructions, security checkpoints, instruction reordering, etc. With the automatic analysis the robustness against certain attacks can be increased while the simulation time is short.

6. CONCLUSIONS

In this paper we have presented a novel methodology for hierarchical security attack simulation at different levels of abstraction. The design flow for power aware smart cards has been outlined and the hierarchical security attack simulation including the attack simulation at the processor specific model level has been described. Results show the impact of faults, resulting from simulated security attacks, on the power profile. Moreover, they show that this method of analyzing the behavior using the power information of the system is suitable to evaluate the fault injection methodology at the higher abstraction levels such as functional level or the intermediate platform model at transaction layer 1 (TL1). Furthermore, system vulnerabilities due to information leaking through the power consumption profile can be detected by using the attack simulation and automatic profile analysis at TL1 in SystemC.

Future work will be done in further evaluation of the methodology of the analysis of the fault's impact on the power consumption.

7. REFERENCES

[1] R. Anderson, M. Kuhn, “Tamper resistance - A cautionary note,” *Second USENIX Workshop on Electronic Commerce*, Nov '96.. 1996. pp. 1-11.

[2] Open SystemC Initiative (OSCI), “SystemC 2.0 Language Reference Manual”, Revision 1.0, www.systemc.org, 2003.

[3] J.-F. Dhem, N. Feyt, “Hardware and software symbiosis helps smart card evalution,” *IEEE Micro*, 21(6):14-25, December 2001.

[4] Y. Gressus, “New methodologies in smart card security design,” *Smart Card Security Conference*, Japan, 2001.

[5] J. J. A. Fournier, S. Moore, H. Li, R. Mullins, G. Taylor, “Security Evaluation of Asynchronous Circuits,” *Cryptographic Hardware and Embedded Systems* - CHES 2003, 5th International Workshop , Sept. 2003, Proceedings, volume 2779-LNCS, pages 137-151. Springer-Verlag, 2003.

[6] U. Neffe, K. Rothbart, C. Steger, R. Weiß, E. Rieger, A. Mühlberger, “System Design Based on Different Levels of Abstraction for Power-Aware Smart Cards,” *Austrochip 2003*, - in: Tagungsband (2003). Page(s): 37 -40.

[7] K. Rothbart, U. Neffe, Ch. Steger, R. Weiss, E. Rieger, A. Muehlberger, “High level fault injection for attack simulation in smart cards,” *Test Symposium, 2004. (ATS 2004). Proceedings of the Thirteenth Asian* , Nov. 2004. Pages:118 - 121.

[8] K. Rothbart, U. Neffe, Ch. Steger, R. Weiss, E. Rieger, A. Muehlberger, “A Smart Card Test Environment Using Multi-Level Fault Injection in SystemC”, *6th IEEE Latin-American Test Workshop (LATW'2005) Digest of Papers*, March-April, 2005, pp. 103-108

[9] MIPS Technologies, *Inc. MIPS32 4KSTM Processor Core Family Software User's Manual*. www.mips.com, 2001.

[10] U. Neffe, K. Rothbart, C. Steger, R. Weiss, E. Rieger, A. Muehlberger, “Energy estimation based on hierarchical bus models for power-aware smart cards,” *Design, Automation and Test in Europe Conference and Exhibition*, 2004. Proceedings ,Volume: 3 , Feb. 2004, pp:300 - 305 Vol.3.

[11] Sun Microsystems, Inc., “Java Card^{TM} virtual machine specification,” http://java.sun.com/products/javacard/, 2001.