On the Portability of Side-Channel Attacks

– An Analysis of the Xilinx Virtex 4 and Virtex 5 Bitstream Encryption Mechanism –

Amir Moradi, Markus Kasper, Christof Paar

Abstract—**This paper is a short summary of our real-world side-channel analysis of the bitstream encryption mechanism provided by Xilinx Virtex FPGAs. This work covers our results analyzing the Virtex 4 and 5 family showing that the encryption mechanism can be completely broken with moderate effort. The presented results provide an overview of a practical real-world analysis and should help practitioners to judge the necessity to implement side-channel countermeasures. We demonstrate sophisticated attacks on off-the-shelf FPGAs that go far beyond schoolbook attacks on 8-bit AES S-boxes. We were able to perform the key extraction by using only the measurements of a single power-up. Access to the key allows cloning and manipulating a design, which has been encrypted to protect the intellectual property and to prevent fraud. As a consequence, the target product faces serious threats like IP theft and more advanced attacks such as reverse engineering or the introduction of hardware Trojans. To the best of our knowledge, this is the first successful attack against the bitstream encryption of Xilinx Virtex 4 and Virtex 5 reported in the open literature.**

I. Introduction

FPGAs (Field Programmable Gate Arrays) are a powerful tool to design products that require hardware performance without having the costs and delays of ASIC (Application Specific Integrated Circiut) development. Furthermore, FPGAs allow in the field updates and are thus way more flexible than ASICs.

During powerup, an SRAM-based FPGA reads its configuration from an external non-volatile memory. The configuration includes all functional design as well as the I/O configuration for the pins and the exact placement and routing of all used components. Copying a configuration to use it for multiple FPGAs, makes all devices behave in exactly the same way. The whole design of an FPGA application is encoded within the configuration file the role of which can be considered similar to the role of software for microcontrollers. On the one hand, this nature provides a means to update the configuration file of an FPGA to adapt its behavior to new requirements or to fix early design flaws. On the other hand this also simplifies copying of a design and thus stealing of IP (Intellectual Property). Today it is even possible to reverse-engineer FPGA configuration files [6] so that the possibility of eavesdropping a bitstream – the name of the configuration data on Xilinx FPGAs – leaves doors wide open for product piracy and IP theft and makes cloning of unprotected FPGA designs easy.

To counter these threats Xilinx as of 2001 [8] implemented an encryption mechanism in many of its recent FPGA series released within the last decade. This mechanism is called bitstream encryption and works in the following way: instead of saving a plain bitstream file within the configuration ROM (Read Only Memory) feeding an FPGA, the designer stores an encrypted bitstream configuration. The encryption – using AES-256 in CBC (Cipher Block Chaining) mode for the discussed FPGAs – is performed in software by the Xilinx ISE development tools. The used key is chosen by the designing engineer and is programmed into the Virtex FPGA. The part of the FGPA memory storing this secret key is battery powered so that the key will immediately be erased on power loss of the battery support. This feature is designed to hinder invasive attacks to recover or reverse engineer a device configuration.

With the known encryption key inside the FPGA and the encrypted bitstream stored within a ROM, products can securely configure the FPGAs as only AES-256 encrypted data passes the channel between ROM and FPGA. The FPGA has a dedicated AES hardware to decrypt the bitstream. This hardware is not accessible for other purposes within the FPGA due to export regulations of cryptography.

Recently a successful side-channel key recovery attack on the bitstream encryption feature of Xilinx Virtex-II pro FPGAs, which employ 3DES as the decryption engine, has been reported in [4]. In this paper we provide an overview of a practical side-channel analysis attack on the bitstream decryption engines of Virtex 4 and Virtex 5 FPGAs. These attacks demonstrate that industrial products in fact require to implement side-channel countermeasures and that side-channel attacks are not a pure academic playground but have a real-world impact on the security of embedded systems.

This summary paper is organized as follows: after this introduction to the topic we discuss the basic elements of the implemented attack. Afterwards, we give a brief overview of the achieved results highlighting attack complexity and real-world feasibility. Finally we provide a short conclusion on the practical impact of our results.

II. Side-Channel Analysis

Today Side-Channel Analysis (SCA) is a mature field in applied security research. Differential side-channel analysis methods have been introduced first by Kocher et al. around 10 years ago [3]. Since then the field has grown rapidly and many new tools and distinguishers for side-channel analysis have been evaluated. In reply to the new threat developed in the scientific literature many countermeasures have been proposed,

A. Moradi, M. Kasper and C. Paar are with the Horst Görtz Institute for IT-Security, Ruhr University Bochum, Germany, e-mail: {moradi, mkasper, cpaar}@crypto.rub.de

implemented and broken. Also, experts from the field of theoretical cryptography recognized side-channel attacks as an important topic seeding a community of researchers working on general leakage resilience and provable security bounds for side-channel countermeasures. Beyond academic purposes, side-channel attacks and reverse engineering have been shown to have real-world impact. Examples are the attacks on NXP's Mifare Classic devices [5], a bouquet of attacks on Microchip's KeeLoq remote keyless entry systems (primary article [2]), and recently also SCA attacks on Mifare DESFire contactless smartcards [7].

The method used in this work is a sophisticated type of Correlation Power Analysis (CPA) as first introduced in [1]. In this method the power consumption of a device is measured while executing a cryptographic algorithm. In addition to the physical power consumption of the analyzed device, also the communication of the device is eavesdropped to get access to the ciphertexts (or plaintexts) that will be (or have been) processed. In our case the ciphertexts, i.e., the encrypted bitstream, is available by eavesdropping the configuration process and the analyzed cryptographic primitive is an AES-256 decryption.

During the analysis itself the known ciphertexts are used to predict an intermediate value processed by AES for each measured power trace. To do so, a fixed hypothesis for the part of the key determining the observed intermediate value is made and applied when calculating the hypothetical intermediate value for each trace. In the next step the hypothetical values are used in a hypothesis test, which distinguishes the key used by the device from wrong key hypotheses. In a CPA attack this distinguisher is Pearson's correlation coefficient estimated by the sample correlation. To apply this distinguisher the predicted intermediate values have to be mapped to hypothetical power consumptions, which will then be compared with the measured power consumption. For hardware designs a reasonable choice to do this is the Hamming distance (HD) model, which counts the number of bits of an intermediate value that are toggled from one clock cycle to the next.

Side-channel analysis attacks follow a divide-and-conquer strategy. That is the key is recovered in small pieces. Typical attacks use subkeys of 8 (AES) or 6 (DES) bits and target S-box outputs.

In our attack we can use a full bitstream as a set of multiple ciphertexts. We performed all our analysis on a Virtex 4 device and afterwards applied the same attacks Virtex 5 devices. The main difference was that the attack on Virtex 5 FPGAs required more power traces to be successful, which is mostly due to a worse signal-to-noise ratio due to a newer process technology (i.e., 65 nm instead of 90 nm). For the attack to work we applied a bandpass filter on our measured traces. Furthermore, we were able to improve our analysis method by removing all phase shifts from the measured traces using an additional FFT preprocessing step.

III. Experimental Results

During one single power-up we measured the power consumption of the decryption of 50 000 (Virtex 4) or 90 000 (Virtex 5) encrypted bitstream blocks (128-bit each). Analyzing the power traces using a known key, we were able to uniquely identify the time instances where the decryption happens. By trial and error we were able to predict the structure of the AES-256 hardware architecture, which was the same for Virtex 4 and Virtex 5. The architecture consists of a full AES round which is evaluated in parallel, and which is repeated 14 times for the encryption of one block. Having a good idea of the used decryption architecture we selected an appropriate intermediate value and power model. As the recovered architecture calculates full AES rounds within a single clock cycle we started by using full 32-bit key hypotheses in our attack.

Using off-the-shelf hardware (no special side-channel evaluation board), we designed an attack that recovers the full secret key by extracting eight sets of each 32 bits. This implies that the attack tested 2^{35} key hypotheses in total, corresponding to (depending on the number of traces used) around 2^{51} hypothetical intermediate values to predict. Using double precision arithmetic, the resulting set of correlation coefficients has a size of 256 Gigabyte. There has not been any real-world side-channel analysis to our knowledge before, which had a comparable complexity. Although this sounds like a tough computational problem, in practice the attack itself could be parallelized to a set of four nVidia Fermi GPUs (Tesla C2070). A full key recovery using 50 000 measurements finishes in 8×39 minutes, i.e., in 6 hours (Virtex 4), and a full recovery on Virtex 5 devices using 90 000 measurements finishes in 8×67 minutes,i.e., about 9 hours. Note that since we have found the time instance when the desired decryption round is executed, we have restricted our attacks to only a single point of the power traces.

These are just first results and we believe that it is possible to further reduce the number of required traces and hypotheses in the future.

IV. Conclusion

IP theft and product piracy are important topics in many industries. Our attacks show, that the IP protection of the analyzed FPGAs can be circumvented. But there are also consequences other than IP theft implied by the insecurity of the bitstream encryption. An attacker cannot only extract and reverse engineer the bitstream, but he might also modify it or create a completely new one, which would be accepted by the device for configuration. This fact is especially sensitive in military applications, but could also have a major impact in other fields as surveillance and Trojan hardware scenarios. Furthermore, an unencrypted bitstream allows to read out secret keys from security modules or to recover classified security primitives. In this work we demonstrated that real-world attacks beyond 8-bit hypotheses are feasible and need to be taken into account when designing a secure system. As of today side-channel countermeasures are mostly employed in high-security devices, such as smartcards for banking or pay-TV applications. Other industries so far mostly avoided the additional costs and efforts and use proved cryptographic primitives without providing countermeasures. We believe this is mainly due to the fact that SCA attacks are still believed to be of academic interest only without having much impact on real-world security. With this attack we provided another case for the "feasibility" of side-channel attacks. Even complex ICs using recent semiconductor technology can be attacked within a reasonable time.

V. Acknowledgements

The authors want to thank Alessandro Barenghi from Politecnico di Milano for providing the programming tools needed to transfer our attacks to the Virtex 5 FPGA family. Furthermore, we want to thank David Oswald for many fruitful discussions and his advices on the Fourier preprocessing.

The work described in this paper has been supported in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II and by the German Federal Ministry of Education and Research BMBF (grant 01IS10026A, Project EXSET).

References

[1] E. Brier, C. Clavier, and F. Olivier. Correlation Power Analysis with a Leakage Model. In *CHES 2004*, volume 3156 of *LNCS*, pages 16–29. Springer, 2004.

[2] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani. On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme. In *CRYPTO*, volume 5157 of *LNCS*, pages 203–220. Springer, 2008.

[3] P. C. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology - CRYPTO '99*, volume 1666 of *LNCS*, pages 388–397. Springer, 1999.

[4] A. Moradi, A. Barenghi, T. Kasper, and C. Paar. On the Vulnerability of FPGA Bitstream Encryption against Power Analysis Attacks – Extracting Keys from Xilinx Virtex-II FPGAs. In *the 18th ACM Conference on Computer and Communications Security - CCS 2011*. ACM, 2011. to appear. A draft version is available in Cryptology ePrint Archive.

[5] K. Nohl, D. Evans, Starbug, and H. Plötz. Reverse-Engineering a Cryptographic RFID Tag. In *USENIX Security Symposium*, pages 185–194. USENIX Association, 2008.

[6] J.-B. Note and É. Rannaud. From the bitstream to the netlist. In M. Hutton and P. Chow, editors, *16th International Symposium on Field Programmable Gate Arrays, FPGA 2008*. ACM, 2008.

[7] D. Oswald and C. Paar. Breaking Mifare DESFire MF3ICD40: Power Analysis and Templates in the Real World. In *CHES 2011*, LNCS. Springer, 2011. to appear.

[8] S. Trimberger. Trusted design in FPGAs. In *the 44th annual Design Automation Conference*, DAC 2007, pages 5–8. ACM, 2007.