Lightning Talk:
The Incredible Shrinking Black Box Model

Patrick Schaumont
Worcester Polytechnic Institute, Worcester, MA 01609

Abstract—A black box model is an assumption on the implementation of a cryptographic primitive to limit the capabilities of the attacker. Black boxes are a useful component in a proof of protocol correctness, but it is not obvious how to securely implement one in hardware. The current state of the art in tamper from open literature shows impressive efficiency and precision in prying open these black boxes. Secure hardware designers are compelled to shrink the black boxes with every new device generation, while making a careful assessment of the need to load, store and handle secrets in hardware.

Index Terms—hardware security, cryptographic engineering, reverse engineering

I. INTRODUCTION

An important corollary from Kerckhoffs’s law, which states that the security of a cryptosystem should only depend on the secrecy of the key, is that the implementation of the cryptosystem must ensure the secrecy of the key. To express this constraint, cryptographers use black box models that turn a cryptosystem’s operations opaque to the outside world. Inside of the black box model, the secret key can be used without any risk to its secrecy. The black box model encompasses the secret key as well as the encryption algorithm that uses the key. The hardware implementation of a black box model thus brings three practical but difficult design problems that must be addressed by the hardware designer: (a) how to insert/ remove a secret into/from the design, (b) how to store a secret into the design, and (c) how make computations that use the secret (such as during encryption). These challenges have received considerable attention in the hardware security community, most often from the offensive side. The objective of this lightning talk is to review some of the recent results in black-box tampering. We first consider static secrets stored as a hardware state, and next consider the implications of secure hardware processing with secrets.

II. SECRETS AT REST

A variety of techniques are used to store secrets into silicon, from straightforward memory cells or registers, to fuses and physically unclonable functions (PUFs). The latter two are useful to support the insertion/removal of non-volatile secrets into the design in a scalable manner, but they are no magic bullet for the black box design. PUFs grow the black-box in size because their noisy nature requires error coding to produce a stable secret key. Furthermore, hardcoded secrets have limited use in cryptosystems that require frequent key-update, for example to ensure forward security and postcompromise security [1], [2]. Practically secure systems use a combination of one hardcoded secret for authentication, and ephemeral secrets for all other information security tasks.

Assuming that a secret key is stored inside of a few register cells in a multi-million-gate ASIC, how difficult is it to extract the secret key? The answer to this question has two parts: (a) one must know where to look into the ASIC, and (b) one must be able to physically read out the content of the registers. Thus, one must solve a reverse engineering challenge and a measurement challenge. The semiconductor manufacturing industry already uses a set of tools to examine the structure and logic state of digital circuits after their manufacturing [3]. These tools, originally developed to study reliability and defects in ICs, can be repurposed by the attacker as powerful probing mechanisms to measure the state of a specific circuit node in a complex design. They require optical or conductive access to the backside or the frontside of the die, and may require special sample preparation such as package decapsulation, die polishing, or drilling holes in the top-layers of the die through high-energy ion beams. These techniques may take time to learn, but the important point is that the hacker community doesn’t have to invent them [4].

The real challenge of extracting secrets at rest comes from the required know-how to make the attack efficient. Module boundaries, which are essential to make sense of design data, are absent at the lowest abstraction level of implementation. However, it is possible to identify word-level structures or FSMs almost automatically [5], [6]. In case of reconfigurable logic, recent tools can reconstruct the netlist from the low-level bitstream [7]. Performing further analysis beyond identifying these very basic structures requires domain knowledge regarding the type of cipher and the manner in which the secret is used. For example, the bitstream decryption engine of an FPGA is active during the configuration process, so identifying the active portion of an FPGA at that time may reveal the location and structure of the cipher [8]. In a more advanced analysis, machine learning is combined with failure analysis to identify the sensitive locations automatically [9].

We thus conclude that a standard IC, even at advanced technology nodes, can only emulate the black-box model to a limited extent. Black-boxes in standard IC technology are not effective against a knowledgeable and determined attacker. A long-term, hardcoded secret in an IC now appears to be bad idea.

This research was supported in part through the National Science Foundation Grant no. 2219810.
III. SECRETS AT WORK

The black box model also includes all the operations that manipulate secrets as part of data transfer, computations and storage. The attacker uses the implementation effects of these operations in a side-channel attack. Side-channel analysis techniques have been refined for a quarter of a century now, and they have become a major obstacle to the realization of the black box model. There are several reasons why side-channel leakage is such a major issue.

First, new sources of side-channel leakage keep on popping up. The classic sources of side-channel leakage such as first-order data-dependency on power and timing, are only the tip of the iceberg. Significant additional leakage originates from complex higher-order interactions in the implementation such as coupling effects on the interconnect [10], or signal-integrity effects on the power grid [11]. These sources of side-channel leakage are hard to understand for the designer, in the first place because the IC design flow considers them as a parasitic second-order effect. Their modeling and simulation is cumbersome, and hence many of these side-channel sources have only been discovered on fielded systems.

Second, the highly hierarchical nature of system design, which strictly isolates design abstraction levels (such as software, ISA, micro-architecture, circuit), makes it extremely difficult to trace the outline of the black-box model. If a secret is manipulated, it is very difficult to predict how long the effects of that manipulation will linger; we learned this lesson the hard way when speculative attacks were first identified [12], [13]. Information flow analysis can help, provided that it captures the information flows at the lowest possible abstraction level [14].

Third, the notion of side-channel leakage itself is fluid and changing. Traditional side-channels are a byproduct of cryptosystem activities, which brings the leakage under control and responsibility of the hardware designer. However, side-channel leakage can also occur from active manipulation by the attacker, such as through thermal laser stimulation or fault injection [15]. Inducing side-channel leakage can be as simple as disabling a random number generator [16].

Finally, side-channel analysis techniques that extract the secret from the leakage, keep on improving. New categories of cryptosystems lead to novel analysis techniques. For example, the long-duration traces of post-quantum cryptography allow specialized horizontal side-channel analysis [17]. And machine learning appears to be a game changer in universally exploiting side-channel leakage [18].

IV. CONCLUSIONS

A feasible long-term solution of a hardware black-box appears increasingly unlikely. Shrinking the black-box in physical size was essential, but it may no longer be enough. Perhaps the next dimension to tackle is to shrink the black box in time, and instead consider a sound cryptography-based strategy to continuously refresh black-box secrets and to even recover the black-box secrecy after a security compromise.

REFERENCES