Trustworthy computing on untrustworthy and Trojan-infected on-chip interconnects

Trustworthy computing on untrustworthy and Trojan-infected on-chip interconnects

Heba Salem
School of Informatics
The University of Edinburgh
Edinburgh, United Kingdom
s1573838@ed.ac.uk

Nigel Topham
School of Informatics
The University of Edinburgh
Edinburgh, United Kingdom
nigel.topham@ed.ac.uk

***Abstract*—This paper introduces a scheme for achieving trustworthy computing on SoCs that use an outsourced AXI interconnect for on-chip communication. This is achieved through component guarding, data tagging, event verification, and consequently responding dynamically to an attack. Experimental results confirm the ability of the proposed scheme to detect HT attacks and respond to them at run-time. The proposed scheme extends the state-of-art in trustworthy computing on untrustworthy components by focusing on the issue of an untrusted on-chip interconnect for the first time, and by developing a scheme that is independent of untrusted third-party IP.**

***Index Terms*—Hardware Trust, Hardware Security, Design for Security**

I. Introduction

Globalization in the IC supply chain raises a particular concern about the malicious modifications that potentially untrusted parties in this chain could introduce in SoCs, *i.e.,* hardware Trojans (HTs). A type of HTs that have gained special attention in literature are HTs embedded in outsourced IPs, as such IPs are heavily relied-on in the SoC industry given the stringent limitations on design overhead and time-to-market. A large number of publications attempt to address the issue of HTs in 3PIPs from the pre-deployment HT detection perspective. However, the extent to which real-life HTs will pass undetected is an open question, as our knowledge of HTs is based on some hypothetical assumptions and most papers evaluate their ideas using research-based benchmarks [1], rather than realistic Trojans. Consequently, to achieve robust protection against HTs, systems should include dynamic countermeasures that are capable of detecting the malicious actions of HTs and responding to them effectively at run-time. Such countermeasures aim to achieve trustworthy computing on untrustworthy components (TCUC). At its core, TCUC involves run-time monitoring combined with proactive dynamic responses to any inferred HT attack.

Run-time monitoring is effectively the last line of defense against HTs and, according to the latest survey on HTs and their countermeasures [2], developing run-time monitoring techniques is a key future research direction for secure computing. To this end, several TCUC techniques have been proposed in the literature; that are based on mechanisms such as task scheduling [3], verifiable computing [4], and the implementation of micro-controlled elements in enforcing security rules [5]. The issues identified in prior TCUC proposals indicate the importance of developing techniques that are: (a) widely applicable in many types of SoCs, (b) impose minimal design and computations overheads, (c) are hard-wired for resilience against software attacks, and most importantly (d) are fully independent from third-party IP vendors. The TCUC scheme proposed in this paper addresses these requirements, and to our knowledge, is the first TCUC work that succeeds in doing so.

II. Threat model

The AXI protocol is one of the most widespread of the AMBA protocols introduced by ARM. Due to their prevalence, AXI interconnects are viewed as standard circuit elements and are often outsourced, raising in turn serious concerns regarding their trustworthiness. The HT scenarios expected from an untrustworthy AXI interconnect and that are targeted in this work are data diversion in which the interconnect diverts packets from their actual destination, data modification where the interconnect manipulates data that passes through it, shadowing attacks in which the interconnect keeps a copy of the data passing through it so that it can later leak it out of the system, illegitimate access to resources by trying to initiate read/ write requests without receiving corresponding requests from the master IPs in the SoC, and finally DNS of the request or data flooding type. The targeted HT scenarios that could be initiated by the untrustworthy 3PIPs through their facilitation of the AXI interconnect or in collusion with it are illegitimate access to a certain slave IP/ address by one of the master IPs, masquerading attacks where one IP poses as another to the interconnect, and shadowing and flooding-based DNS attacks that revolve around a slave IP using the interconnect to transmit some data without it being involved in a corresponding on-going transaction.

III. The proposed TCUC scheme

The HT scenarios presented in section II indicate that the root cause behind the potential success of any of them is the capability of an untrustworthy AXI interconnect or 3PIP to either modify a transaction as it passes through the interconnect or to initiate an unexpected transaction. The absence of any limitations on their actions and interactions

represents a vulnerability that must be mitigated at run-time by imposing constraints that restrict the actions of a 3PIP to only those that satisfy its functionality and comply with its specification. The key to our proposal is an appropriate set of run-time constraints on the interconnect and the 3PIPs that can provide the security guarantees that would counter the presented HT scenarios and achieve TCUC. The proposed TCUC scheme is based on combining the guiding principles of *component guarding*, *event verification*, and *data tagging*. These guiding principles are summarized as:

- Component guarding: involves monitoring, logging, and keeping track of the progress of every transaction that passes through the interface of the untrustworthy 3PIP.
- Event verification: involves ensuring that the events occurring at the interface of the untrustworthy 3PIP are expected and within its functional and access boundaries.
- Data tagging: involves using tags and IDs to identify the transmitted data and confirm its integrity.

These guiding principles are imposed by security elements that act as wrappers around the IPs in the SoC. These wrappers are applied to the IPs in the system by the SoC integrator at build time, and require no internal knowledge of the IPs themselves. Each wrapper has a unique ID that identifies both the wrapper and the data issued by that wrapper; an important measure in preventing masquerading attacks. The wrappers serve as monitoring and HT-action prevention elements and they have full observability over the signals and events that occur at the interfaces of the wrapped IPs.

An example of a SoC with n master IPs and m slave IPs is shown in fig. 1. In this figure the wrapper of the AXI interconnect is denoted as W(AXI) whereas the wrappers of the master IP M_i and slave IP S_j are denoted $W(M_i)$ and $W(S_j)$, respectively. The wrapper of the AXI interconnect contains master and slave interface units $U(M_i)$ and $U(S_j)$ for each master and slave respectively. Each interface unit implements state machines to monitor and verify the signals for the five AXI channel. Additionally, the AXI wrapper contains a collection of small memories to log key information related to each transaction that passes through the interconnect. The transaction logs are denoted TR_log, whereas DATA_log denotes the logs that track data transfers. The logs are arranged to allow simultaneous access from master and slave interface units, thereby allowing transactions to proceed through the AXI network without any reduction in throughput. The wrapper of a master IP comprises an access control unit and a data tag generation unit for generating tags for the data beats sent by the master IP when in write data mode. The wrappers of slave IPs consist of data tag generation units that are associated with the read data. Each tag is specific to the data beat and the ID of the source IP for that data.

IV. Results and Functional Verification

Wrappers for our proposed TCUC mechanism were written in Verilog and integrated in a modelled SoC based on a combination of open-source and Xilinx IP blocks. The design was simulated under both normal conditions and malicious

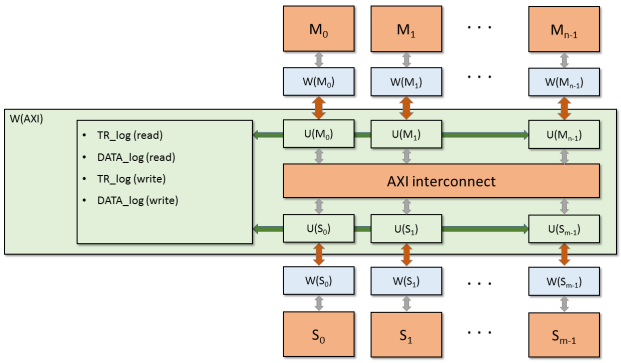

Fig. 1. Deployment of proposed TCUC IP wrappers in a SoC

conditions where HT scenarios presented in section II were modeled and induced in the SoC. The simulated results show that in both cases of operation, the wrappers add a fixed latency of three clock cycles to each of the channels in the AXI read and AXI write interfaces (except the write response channel, in which the added latency is two clock cycles). Additionally, all of the induced HT scenarios were successfully detected by the proposed wrappers, and the corresponding transaction in each case was blocked.

V. Conclusion

This paper presents the first TCUC technique that targets on-chip interconnects. The proposed technique extends the state-of-art in the field through its independence from untrusted IP vendors, its low computational overhead, and its hardwired implementation. By developing a fully hardwired technique we minimize the threat surface, as it cannot be attacked from the software level. Additionally, the proposed technique is somewhat universal as its guiding principles could be applied to any on-chip interconnect. The wrappers developed to test the principles of the scheme can be configured and deployed with comparatively low effort, mainly requiring an SoC integrator to specify the number of master and slave IPs, the allowed address range for each master IP, and the size of the transaction logs.

References

[1] H. Salmani, M. Tehranipoor and R. Karri, "On design vulnerability analysis and trust benchmarks development," in 2013 IEEE 31st International Conference on Computer Design (ICCD), Asheville, NC, 2013, pp. 471-474.

[2] X. Mingfu, G. Chongyan, L. Weiqiang, Y. Shichao and M. O'Neill, "Ten years of hardware Trojans: a survey from the attacker's perspective," IET Computers and Digital Techniques,vol.14, pp.231–246, November 2020.

[3] D. McIntyre, F. Wolff, C. Papachristou and S. Bhunia, "Trustworthy computing in a multi-core system using distributed scheduling," in 2010 IEEE 16th International On-Line Testing Symposium, Corfu, 2010, pp. 211-213.

[4] R. S. Wahby, M. Howald, S. Garg, A. Shelat and M. Walfish, "Verifiable ASICs," in 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, 2016, pp. 759-778.

[5] A. Basak, S. Bhunia and S. Ray, "A flexible architecture for systematic implementation of SoC security policies," in 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, 2015, pp. 536-543.