Attack on Memory Encryption in MPSoCs using HT-infected AES core

Nilanjana Das, Mattis Hasler, Friedrich Pauls, Yogesh Verma, Sebastian Haas
Barkhausen Institut, Dresden, Germany
{first name.last name}@barkhauseninstitut.org

***Abstract*—In this work, we explore the limitations of isolation-based architecture and Trusted Communication Units (TCUs) in securing Multi-Processor System-on-Chip (MPSoC) platforms against hardware attacks. Despite these security measures, the presence of a Hardware Trojan (HT) can introduce a backdoor capable of leaking sensitive data from encrypted memory. The proposed HT operates independently of the TCU, allowing for unauthorized access without detection. The HT is incorporated into an AES accelerator to leak the secret key. The impact of this threat is validated through silicon implementation and a tape-out in 22nm FD-SOI. The area and power overhead are too negligible to detect the inserted HT.**

***Index terms* – MPSoCs, Advanced Encryption Standard (AES), Hardware Trojan (HT), memory encryption, Isolation.**

I. Introduction

The Internet of Things (IoT) is a crucial factor in digitization in many domains, such as industry automation, avionics, and healthcare services. In tile-based MPSoCs, each tile contains IP hardware cores (e.g., a single general-purpose processor, memory, hardware accelerator) or a cluster of IP hardware cores (several processors and shared memories communicating through a bus). In order to increase the performance of the MPSoCs, a memory hierarchy, which includes several cache levels and DRAM, is introduced [1].

It is essential to consider the security factor of an MPSoC from both hardware and software sides. Moreover, modern IC design often involves IP cores supplied by untrusted third-party (3PIP) vendors such as outsourced design, test services, as well as electronic design automation (EDA) [2] software tools.

Security is a prime factor in building trustworthy SoCs. One well-known threat model is to harm external memory by leaking sensitive data. The authors in [3], [4] find that the bus between the SoC chip and the on-chip memory is untrusted and is susceptible to physical attacks. The encryption of data values using algorithms such as the Advanced Encryption Standard (AES) or Data Encryption Standard (DES) guarantees data confidentiality [3]. Therefore, a cryptographic accelerator such as an AES accelerator is essential for memory encryption.

The authors in [1] introduce an HW/OS co-designed platform that supports a microkernel-based OS called M3, where each process and OS service is assigned to its own processor. Isolation is achieved by deploying the TCU.

Our work introduces an Hardware Trojan (HT) attack model to the isolation-based tiled MPSoCs. The attack targets the AES accelerator by inserting an HT into it. The intended use case scenario is that one application will write into the external memory (DRAM). These data will be encrypted before being written into the memory without knowing the secret key. Another application in a different tile will be able to read the raw encrypted data from memory with the help of a malicious attack by eventually leaking the secret key, as shown in Fig. 1(a). As a real-life scenario, the attack model can harm memory encryption by eventually leaking sensitive data. The proposed work considers a malicious application that can leak the secret key of the AES accelerator using the HT and can access sensitive data in memory. The implementation of HT-incorporated AES in silicon on the MPSoCs is the sole focus of this effort. Future research will address this aspect of HT detection techniques.

In summary, the contributions of the paper are as follows: 1) Insertion of an HT incorporated AES accelerator in a silicon implementation of an MPSoCs which utilizes the TCU from [1] and supports the described HW/OS security concept from [5]. 2) It is shown that the HT can leak the secret key of the AES accelerator without being suspicious to the core security component TCU. 3) Lastly, the attack model's real-world application to compromise memory encryption on the MPSoCs platform, despite the existence of isolation and TCU approaches is examined.

The paper is organized as follows. Section II explains the related work. Section III explains the implementation of the HT model in an AES accelerator considering the accelerator support module in isolation-based MPSoCs. Section IV provides the experimental results, followed by a conclusion in Section V.

II. Related Work

This section reviews the state-of-the-art architectures which incorporate isolation between applications and control access to other resources (e.g., memory, accelerators, I/O devices), which is achieved with security features integrated in general-purpose processors. A dedicated hardware component, which checks all requests initiated by a tile, referred as the Memory Protection Unit (MPU), has been explored in the literature. The works in [6] introduce MPUs as hardware firewalls in each NoC interface of a tiled MPSoCs. Similarly, this work [1] improved these MPUs by implementing a framework that enforces access control policies not only for memory but also for various shared peripherals. For example, NoC-MPU [7] is a memory protection unit configured by trust agents—privileged

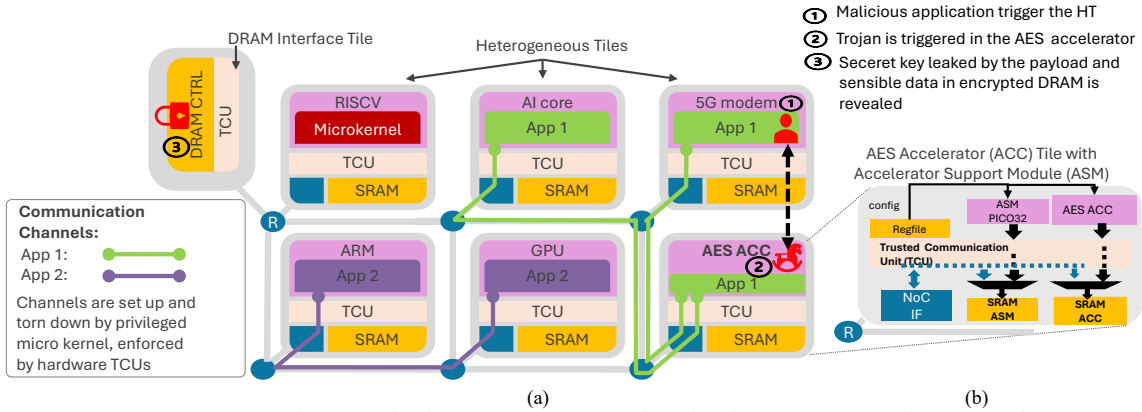


Fig. 1: a) An example attack scenario in memory encryption in the general architecture of a secure HW/OS platform, b)Architecture of the Accelerator Support Module tile with the AES accelerator.

software that establishes and maintains communication between tiles.

For implementation of the proposed attack model the framework presented in [8] is considered. In M3, TCUs already enforce isolation between tiles. The chip platform [8] is based on a tiled hardware architecture, where multiple logically separated tiles are interconnected by a network-on-chip (NoC). Only the OS kernel on a dedicated tile can configure TCUs by setting communication capabilities. Before any communication channel is established, the TCU verifies whether the relevant capabilities are configured; if not, access is denied.

The real-time memory encryption is a strong protection, and it can also prevent another attack, such as memory bus snooping [9]. Similarly, [10] presented and evaluated a process-memory bus encryption technique for embedded systems that requires no changes to applications or hardware. The encryption of values using the AES or Triple DES algorithms before off-chip data transfers guarantees data confidentiality [11].

Hardware security is crucial for all hardware platforms, MPSoCs are not out of scope for HT-related attacks. Numerous research works focus on this area by implementing HTs in different locations of MPSoCs and providing mitigation techniques to evade those attack models [12].

This work incorporates an AES accelerator in the silicon implementation of an MPSoCs named M24. The implemented AES accelerator contains a Hardware Trojan (HT). The work proposes that HT can work against the isolation approach and harm memory encryption in real-life chip settings. The main aim is to highlight the adverse effect of M3-based hardware-software co-design systems. The M3 system is secured by isolation and TCU, which only gives access to the application with a valid request confirmed by the TCU. It is evident that an external memory request from any application is a valid request concerning the TCU, and it will grant the request to the application when it requests memory access. This work shows that the HT in the AES accelerator can leak the secret key and harm the whole system by affecting memory encryption. The activation of the proposed HT is not dependent on the TCU; rather, it depends on data (plain text). The HT can leak the secret key after a successful payload operation and can easily sneak into encrypted memory.

III. The Proposed Work

A. Description of isolation based MPSoC and TCU

The objective of isolating hardware and software components by default can be achieved by the MPSoC design presented by the authors in [1]. This architecture allows individual applications and software components of a microkernel-based operating system to be mapped onto separate tiles, thereby establishing a substrate for trustworthy computing as shown in Fig. 1(a). A key element of this design is the TCU, which connects each component to the NoC, thereby enabling secure communication between all components. Only the OS kernel, running on a dedicated tile, can configure TCUs by setting communication capabilities. These capabilities are stored in endpoint registers. Before any communication channel is established, the TCU verifies whether the relevant capabilities are configured. Otherwise, access is denied. OS features such as virtual memory and context-switching enhance the performance of the software but also require support from the TCU.

B. Accelerator support module tile

To ease this integration challenge, a modular ASM tile is incorporated that reduces the barrier for system integrators to adopt a secure platform [13] as shown in Fig. 1(b). The ASM tile employs the AES accelerator (ACC), which has access to local SRAM via a standard memory interface of up to 128-bit data width. Typically, ACCs are configured via configuration registers or memory-mapped IO. Both variants can be handled by the 64-bit configuration interface provided by the TCU. In addition, it can access the ACC configuration registers

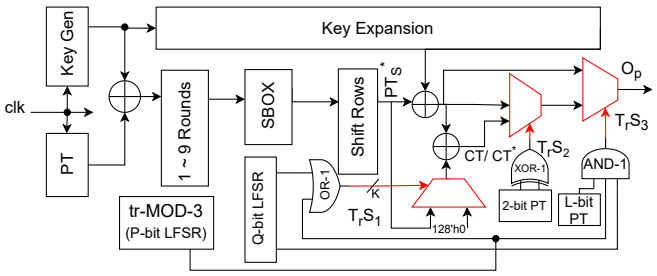


Fig. 2: HT inserted in AES core

TABLE I: M24 area utilization summary

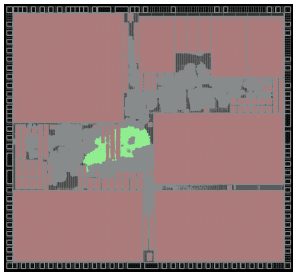
Module	Area (mm^2)
Total chip	11.24
SRAM	3.84
3xRISC-V Rocket tiles	4.08
1xRISC-V BOOM tile	1.81
4xC2C tiles	0.67
NoC	0.07

and possibly interrupt signals via the configuration interface provided by the TCU which enforces all security properties and allows IP integration with minimal security assumptions.

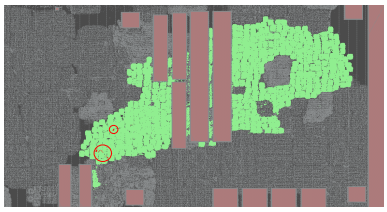
C. Hardware Trojan model in AES accelerator

In this section, we provide details about the design and implementation of the proposed HT model. To activate the HT, we need to activate all the Trigger Signals (TSs) at the same time to satisfy the malicious intention of the adversary. This allows the trigger to function maliciously in order to leak secret information. Note that we assume trigger based HT is implemented in the Register Transfer Level (RTL) design phase, and the Trigger signals are frequently activated instead of rarely activated. The activation of HT depends on three TSs [14] where T_rS_1 depends on the OR-1 gate, Q-bit LFSR, and tr-MOD-3, T_rS_2 only depends on particular 2 bits of plain text, and T_rS_3 depends on AND-1 gate, Q-bit LFSR, tr-MOD-3, and plain text respectively. The payload segment of this HT circuit consists of 2 to 1 multiplexers and a 2-input XOR gate, and they are activated when both the T_rS_1, T_rS_2, and T_rS_3 are activated. This payload segment results in the expanded key of the tenth round used in that AES core. The original key can be retrieved from the key inversion algorithm. The activation probability of these trigger signals can be derived as follows. The activation probability of T_rS_1, T_rS_2, and T_rS_3 are $\frac{(2^{K-1})\times(2^{Q-K})}{(2^Q-1)}+\frac{1}{3\times(2^P-1)}$, $\frac{1}{2}$, and $\frac{1}{(2^Q-1)}\times\frac{1}{3\times(2^P-1)}\times\frac{1}{2^L}$ respectively. If the values of P, Q, K, and L are set to 3, 5, 3 and 16, then the activation probability of T_rS_1, T_rS_2 and T_rS_3 are 0.92, 0.5 and 2.34×10^{-8}, respectively. The key inversion algorithm can retrieve the secret key ($K_0(0,3)$) from the expanded key ($K_{10}(40,43)$) of 10^{th} round.

In a memory encryption scenario, an adversary in an application can activate the HT in the AES accelerator. Fig. 1(a) illustrates such a case, where the malicious application can activate the trigger condition of the HT in the AES accelerator. In the second step, the payload is activated, by which the application gets the knowledge of the secret key. Afterward, it can use the secret key to reveal sensitive data stored in the encrypted memory. The work manifests that when the HT is not dependent on the TCU for its activation, then it can get activated without being suspicious of TCU. This situation will lead to a possible attack model scenario where HT can affect the MPSoCs, though security primitives such as isolation and TCUs are included.



(a) M24 with AES highlighted



(b) AES with HT marked in circle.

Fig. 3: Implemented Trojan infected AES layout in 22nm technology.

IV. Experimental Results

To evaluate the area and power overhead of the AES accelerator, we implemented and fabricated an instance of an M3 architecture in a 22 nm FDSoI GlobalFoundries process, measured under typical conditions (25 °C, 0.8V). The chip was engineered using a comprehensive suite of EDA tools: Cadence Genus for synthesis, Innovus for place-and-route, Tempus for static timing analysis, and Calibre for physical verification. The architecture consists of 12 tiles: 4xRISC-V tiles (3xRocket, 1xBOOM), 4xChip-2-Chip (C2C) IO tiles, a periphery IO tile, 2xASM Tiles, a DRAM Tile, and a 2x2 starmesh NoC (bandwidth 16 bytes/cycle). All other components are synthesized at 100MHz. The total chip area with respect to 3xRISC-V Rocket Tiles, 1xRISC-V Boom Tile, 4xC2C tiles, and NoC are noted in Table I.

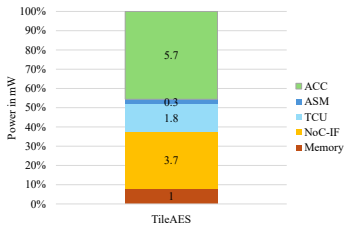
The ASM tile for the AES accelerator have a memory configuration of 4kb ACC SRAM, and 64 kb/16 kb ASM SRAM for code/data for the ASM PicoRV32 protocol core. In this work, we focus on the AES core for a conservative estimate of the area overhead. About 42% of ASM area is used for memory, 11% for the TCU, and only 1% for the ASM PicoRV32 core as in Fig. 4b.

Fig. 3a shows the AES layout in the full chip marked in green. HT is marked in red circle as shown in Fig. 3b. The tile AES requires ACC memory, ASM register file, ASM memory, ACC Multiplexer and ASM Multiplexer for the implementation. The number of clock cycles required to complete the encryption operation is approximately 22 cycles. The total power consumption of the TileAES is 12.8 mW, where memory (ACC memory, ASM memory), NoC-IF, TCU, ASM, and ACC consumes 1 mW, 3.7 mW, 1.8 mW, 0.3 mW and, 5.7 mW respectively as per Fig.4a. Similarly, area consumption for memory, NoC-IF, TCU, ASM, and ACC are 0.144, 0.141, 0.037, 0.005, and 0.094 mm^2 respectively as in Fig. 4b.

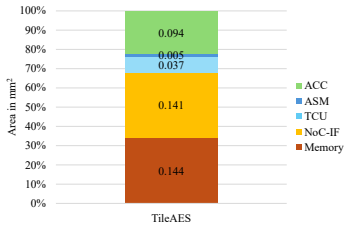
TABLE II: Variation of Resource Utilization of AES core considering with out HT and with HT

Resource	AES core without HT	AES core with HT P = 3, Q =3
Area	0.094 mm^2	0.11 %
Power	5.7 mW	0.07 %

Table II reports the area and total power utilization of AES core without HT and increased resource overhead in percentage with HT. The area utilization for AES with HT is increased up to 0.11% compared to the original circuit for inserted LFSRs. Similarly, the power overhead is increased by 0.07% compared to the original circuit. These area and power overheads are too negligible to detect the inserted HT.



(a) Power Consumption for TileAES



(b) Area Consumption for TileAES

Fig. 4: Implemented Trojan infected AES layout in 22nm technology.

V. Conclusion and Future Work

This research work is mainly focused on the silicon implementation of the AES accelerator in the MPSoC platform. We integrated the HT with the AES accelerator to determine its vulnerability. It is proposed that when the activation condition of the HT is TCU-independent, it can be activated without TCU's permission and harm the MPSoC platform. Hence, the isolation and TCU-based MPSoC are vulnerable to these kinds of HTs. The attack model is applicable to harm DRAM encryption. In future work, we aim to include other security factors along with TCU and isolation so that this kind of HT can be avoided in the MPSoC platform.

VI. Acknowledgment

This research is funded by the European Union's Horizon Europe research and innovation program under grant agreement No. 101094218 (CYMEDSEC) and No. 101092598 (COREnext). It is also financed on the basis of the budget passed by the Saxon State Parliament in Germany.

References

[1] Sebastian Haas and Nils Asmussen. A trusted communication unit for secure tiled hardware architectures. In *2022 29th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 1–4. IEEE, 2022.

[2] Swarup Bhunia, Michael S Hsiao, Mainak Banga, and Seetharam Narasimhan. Hardware trojan attacks: Threat analysis and countermeasures. *Proceedings of the IEEE*, 102(8):1229–1247, 2014.

[3] Zhenglin Liu, Qingchun Zhu, Dongfang Li, and Xuecheng Zou. Off-chip memory encryption and integrity protection based on aes-gcm in embedded systems. *IEEE Design & Test*, 30(5):54–62, 2013.

[4] Johanna Sepulveda, Mathieu Gross, Andreas Zankl, and Georg Sigl. Exploiting bus communication to improve cache attacks on systems-on-chips. In *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 284–289. IEEE, 2017.

[5] Friedrich Pauls, Sebastian Haas, Stefan Köpsell, Michael Roitzsch, Nils Asmussen, and Gerhard Fettweis. On trustworthy scalable hardware/software platform design. In *Smart Systems Integration Conference and Exhibition (SSI)*, 2022.

[6] Benjamin Tan, Morteza Biglari-Abhari, and Zoran Salcic. A system-level security approach for heterogeneous mpsocs. In *2016 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 74–81. IEEE, 2016.

[7] Joël Porquet, Alain Greiner, and Christian Schwarz. Noc-mpu: A secure architecture for flexible co-hosting on shared memory mpsocs. In *2011 Design, Automation & Test in Europe*, pages 1–4. IEEE, 2011.

[8] Sebastian Haas, Christopher Dunkel, Friedrich Pauls, Mattis Hasler, and Yogesh Verma. Trustworthy silicon: An mpsoc for a secure operating system. In *2024 IEEE Nordic Circuits and Systems Conference (NorCAS)*, pages 1–7. IEEE, 2024.

[9] Patrick Colp, Jiawen Zhang, James Gleeson, Sahil Suneja, Eyal De Lara, Himanshu Raj, Stefan Saroiu, and Alec Wolman. Protecting data on smartphones and tablets from memory attacks. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 177–189, 2015.

[10] Xi Chen, Robert P Dick, and Alok Choudhary. Operating system controlled processor-memory bus encryption. In *Proceedings of the conference on Design, automation and test in Europe*, pages 1154–1159, 2008.

[11] Romain Vaslin. 'hardware core for off-chip memory security management in embedded systems. *European University of Brittany*, 2008.

[12] Nilanjana Das, Friedrich Pauls, Mattis Hasler, Sebastian Haas, and Nils Asmussen. Hardware attack models in tiled chip multi-core processors: A survey. In *2024 IEEE 17th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC)*, pages 215–222, 2024.

[13] Friedrich Pauls, Sebastian Haas, and Yogesh Verma. Integration of ip-cores for the m 3 architecture with low area overhead: Accelerator support module. In *2024 21st International SoC Design Conference (ISOCC)*, pages 340–341. IEEE, 2024.

[14] Nilanjana Das, Mattis Hasler, Friedrich Pauls, and Sebastian Haas. A multi-dimensional hardware trojan design platform to enhance hardware security. *IEEE Embedded Systems Letters*, 2024.