Optimizing TLS Cryptographic Operations on RISC-V SoC with OpenTitan RoT

Alberto Musa[1], Emanuele Parisi[2], Luca Barbierato[3],
Edoardo Patti[3], Andrea Bartolini[1], Andrea Acquaviva[1], Francesco Barchi[1]

[1]Università di Bologna, Via Zamboni 33, 40126 Bologna, Italy - {name.surname}@unibo.it
[2]2Barcelona Supercomputing Center - Barcelona, Spain
[3]Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Abstract

This work presents a preliminary evaluation of a cryptographic software stack leveraging OpenTitan as a hardware security module within a RISC-V-based system-on-chip. The current implementation supports the `TLS_RSA_WITH_AES_256_CBC_SHA256` *cipher suite, integrating hardware-accelerated cryptographic operations to enhance security and performance. Through detailed benchmarking, we demonstrate up to 82x speedup for AES-256-CBC and 39x for SHA-256 on larger payload sizes compared to software-only implementations.*

Introduction

Secure communication is crucial in modern embedded systems, particularly given their resource constraints [1, 2]. This work focuses on improving Transport Layer Security (TLS) in a System-on-Chip (SoC) [1] designed for embedded devices that integrate an application-grade processor, CVA6 [2], with OpenTitan, a hardware Root-of-Trust (RoT) [3]. OpenTitan accelerates cryptographic operations, including message digests, symmetric encryption, and asymmetric cryptography, by offloading them to dedicated hardware, enhancing performance while ensuring secure and isolated execution [3, 4]. The goal of this work is to delegate TLS operations to OpenTitan, utilizing its hardware accelerators while maintaining a flexible software stack supporting future cipher suites. The following sections describe the system architecture and software stack, along with performance results showcasing execution time improvements for cryptographic operations.

System Design

The architecture used in this study, shown in Figure 1, is a secure RISC-V-based SoC designed for embedded applications with strong protection against hardware attacks. At its core, the system features a 64-bit CVA6 processor, responsible for system management, peripheral communication, and capable of booting Linux. It is complemented by a cluster of eight CV32E40 cores optimized for computation. OpenTitan is integrated as the hardware RoT, establishing a secure foundation for cryptographic operations and protecting sensitive data. OpenTitan serves as a secure subsystem, providing secure operations and cryptographic acceleration. It includes hardware accelerators for cryptographic operations such as RSA for key exchange, SHA for integrity, and AES for encryption, which are essential

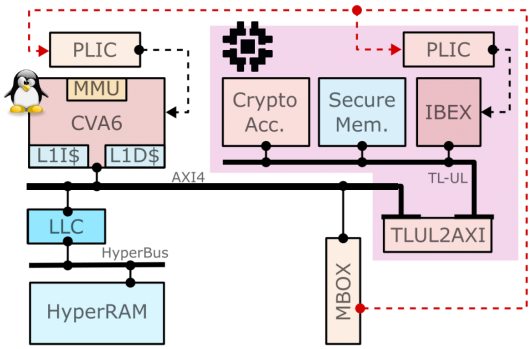

Figure 1: *System architecture of the secure RISC-V-based SoC, integrating CVA6 processor and OpenTitan RoT.*

for various secure communication protocols, including TLS. The memory hierarchy of the system is designed to optimize both performance and security. It includes private L1 caches for the CVA6 processor, ensuring fast access to frequently used data and instructions. A Last-Level Cache (LLC) further enhances memory access efficiency, providing a high-bandwidth cache that benefits both the CVA6 processor and the OpenTitan subsystem. Additionally, the system incorporates 32 MiB of off-chip HyperRAM, serving as the main memory and offering sufficient capacity for handling larger operations and data processing tasks. OpenTitan operates as the master on the bus, granting it full control over memory access across the system. A MailBox system enables communication between the CVA6 Application Core and OpenTitan. This system includes a shared memory region for data exchange, as well as two specialized registers (`Doorbell` and `Completion`) connected to the interrupt controllers of Ibex and CVA6, respectively. This setup ensures efficient asynchronous communication, facilitating the coordination and transfer of commands and data between the CVA6 processor and OpenTitan.

Software Integration

The software stack developed for the target SoC is structured across two main sections: one on the application processor (CVA6) and the other on OpenTitan. On the CVA6 side, the OpenSSL library serves as the entry point and is extended through an OpenSSL Engine, which offloads cryptographic operations to OpenTitan, leveraging hardware acceleration. A Linux driver acts as an intermediary, enabling communication between the OS and hardware components, such as the mailbox and OpenTitan. On OpenTitan, the firmware handles cryptographic operations like SHA-256, AES, and RSA, as requested by the CVA6 processor. The firmware ensures that sensitive operations are executed securely by directly programming the accelerators to perform the required operation using secrets stored in the Secure Memory. A key challenge arises from the fact that OpenTitan lacks an MMU, necessitating the implementation of a manual address management system in the Linux driver to handle virtual-to-physical address translation, ensuring reliability and minimizing the risk of data corruption. However, additional operations for address translation introduce performance overhead. Another challenge is managing the cache asymmetry between CVA6 and OpenTitan, which requires efficient data synchronization. To address these challenges, the communication protocol ensures proper synchronization and formatting of data transfer and cryptographic tasks between CVA6 and OpenTitan via the mailbox. Critical information, such as memory addresses and cryptographic parameters, is managed by a data structure that coordinates the secure exchange of data across the system. This architecture strikes a balance between security and efficiency, optimizing resource usage while overcoming hardware and software constraints.

Performance Evaluation

Our analysis focuses on the core cryptographic algorithms, AES-256-CBC, SHA-256, and RSA, which are required on the TLS cipher suite currently supported (`TLS_RSA_WITH_AES_256_CBC_SHA256`). These algorithms were compared in terms of hardware-accelerated execution on OpenTitan versus an OpenSSL software-only implementation on the CVA6 core, with performance also evaluated against the theoretical limits of the OpenTitan accelerators. Testing was conducted on an Xilinx Virtex UltraScale+ VCU118 FPGA prototype of a RISC-V-based SoC, with Linux running on the CVA6 and OpenTitan serving as the hardware RoT. The evaluation measured the impact of varying payload sizes on throughput and latency. Results summarized in Table 1 clearly demonstrate the performance advantages of OpenTitan's hardware acceleration. For instance, SHA-256 achieved a throughput increase of up to 39.8x at a 16 KB payload size compared to the software-only implementation, while AES-256-CBC reached a speedup of 82.45x at the same payload size. RSA 1024-bit signing achieved a 1.4x speedup, while verification remained unchanged (1.0x) due to computational constraints. Despite the improvements from integrating OpenTitan accelerators, their full potential is not yet fully utilized, especially when looking at their theoretical performance limits. AES and SHA-256 performance varies with payload size. For small payloads, OpenSSL outperforms the hardware-accelerated solution, as the memory management overhead is not compensated by the relatively low amount of data to process. However, the small payloads used in the benchmarks do not reflect real-world scenarios like TLS, where larger payloads are more common. The overhead from memory management limits overall efficiency. In the future, the focus will be on reducing this overhead, optimizing communication, and introducing specialized hardware to improve the interaction between CVA6 and OpenTitan. Additionally, we will explore better utilization of OpenTitan accelerators and extend support for cryptographic operations required by TLS cipher suites.

Table 1: *Results comparison for different payload sizes.*

Payload [B]	Throughput [KiB/s]		Speedup	Theoretical
		SHA-256		
16	43	12	0.29x	0.0 %
32	87	31	0.36x	0.1 %
64	118	55	0.46x	0.2 %
128	178	116	0.65x	0.4 %
256	238	186	0.78x	0.6 %
512	288	382	1.33x	1.3 %
1024	320	537	1.68x	1.8 %
2048	341	1300	3.82x	4.4 %
4096	352	1850	5.25x	6.3 %
16384	357	14222	39.80x	48.5 %
		AES-256-CBC		
16	216	52	0.24x	0.1 %
32	239	140	0.59x	0.4 %
64	250	191	0.76x	0.5 %
128	260	350	1.35x	0.9 %
256	264	402	1.52x	1.0 %
512	265	578	2.18x	1.5 %
1024	267	3840	14.38x	9.8 %
2048	268	4436	16.57x	11.4 %
4096	266	4338	16.30x	11.1 %
16384	259	21333	82.45x	54.6 %

References

[1] Maicol Ciani et al. "Cyber Security aboard Micro Aerial Vehicles: An OpenTitan-based Visual Communication Use Case". In: *ISCAS 2023*, pp. 1–5.

[2] F. Zaruba et al. "The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology". In: *IEEE VLSI* (Nov. 2019).

[3] lowRISC CIC. *OpenTitan Official Documentation*. `https://opentitan.org/book/doc/introduction.html`. 2019.

[4] Alberto Musa et al. "TitanSSL: Towards Accelerating OpenSSL in a Full RISC-V Architecture Using OpenTitan Root-of-Trust". In: *SafeComp 2024*, pp. 169–183.