International Journal of Communication and Information Technology

E-ISSN: 2707-6628
P-ISSN: 2707-661X
Impact Factor (RJIF): 5.56
www.computersciencejournals.com/ijcit
IJCIT 2025; 6(2): 210-213
Received: 02-11-2025
Accepted: 05-12-2025

Kalisetty Sreeja
Assistant Professor, Department of ECE, NRI Institute of Technology, Visadala, Guntur, Karnataka, India

Dr. Saidaiah Bandi
Professor, Department of ECE, NRI Institute of Technology, Visadala, Guntur, Karnataka, India

Corresponding Author:
Kalisetty Sreeja
Assistant Professor, Department of ECE, NRI Institute of Technology, Visadala, Guntur, Karnataka, India

Design of an area efficient RISC-V SoC for En/Decryption acceleration for Homomorphic Encryption

Kalisetty Sreeja and Saidaiah Bandi

DOI: https://doi.org/

Abstract

Edge devices often connect to the cloud these days so they can use its storage and processing power. This brings up concerns about the safety and privacy of user data. Homomorphic encryption (HE) is a good way to protect data privacy because it lets you do any kind of elegant computation on encrypted data without ever needing to decrypt it. There have been a lot of attempts to make HE computations in the cloud faster, but less attention has been paid to the methods of converting messages to ciphertext and ciphertext to messages on the edge. This work profiles the edge-side conversion procedures, and our analysis shows that the encryption, decryption, and error sampling activities are the main problems that slow down the conversion process. To get around these problems, we present RISE, a RISCV SoC that uses less space and energy. RISE uses a lightweight and effective pseudorandom number generator core along with fast sampling methods to speed up the error sampling processes. The number theoretic transform operation is the main bottleneck in the encryption and decryption processes. RISE speeds up these processes by using scalable, data-level parallelism. Also, RISE uses strategies like memory reuse and data reordering to use as little on-chip memory as possible. It also saves space by using a single en/decryption datapath. We use a full RTL design with a RISC-V processor that is connected to our accelerator to test RISE. Our research shows that using RISE instead of just the RISC-V processor makes converting messages to ciphertext and ciphertext to messages much more energy-efficient, by up to 6191.19× and 2481.44×, respectively

Keywords: Homomorphic Encryption, CKKS Scheme, Privacy-preserving Computing, Edge-side Operations, RISC-V, Hardware Acceleration

1. Introduction

With the rapid proliferation of Internet of Things (IoT) devices and the growing reliance on edge computing, data privacy and security have become critical concerns. In many applications, sensitive data generated at the edge—such as medical records, industrial telemetry, or personal user behavior—must be processed without compromising confidentiality. Homomorphic encryption (HE) is a groundbreaking cryptographic technique that allows computations to be performed directly on encrypted data, enabling secure analytics and decision-making without revealing the underlying plaintext.

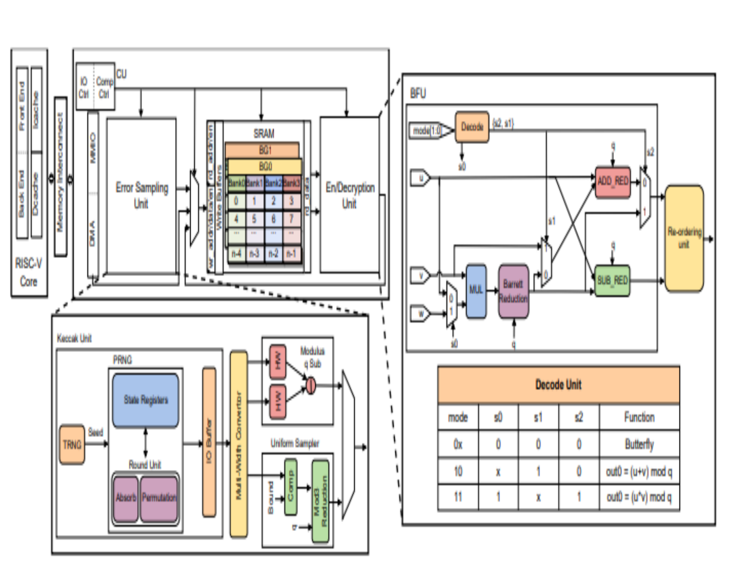
Despite its promise, the practical adoption of homomorphic encryption remains limited due to its high computational overhead. Operations such as modular multiplication, polynomial arithmetic, and number theoretic transforms are intensive and demand considerable processing resources and memory bandwidth. These challenges are particularly pronounced on edge devices, which are often constrained by strict power and area limitations. Traditional microprocessors and general-purpose hardware are not optimized for the specific computational patterns of homomorphic encryption. While software libraries have been developed to support HE on CPUs and GPUs, their energy consumption and latency make them unsuitable for real-time, on-device processing in edge scenarios. This motivates the need for specialized hardware that can deliver both performance and energy efficiency.

RISC-V, with its open-source and modular instruction set architecture, presents an ideal platform for domain-specific acceleration. Its flexibility allows the integration of custom instruction extensions and hardware modules tailored to accelerate specific workloads. In this work, we propose a RISC-V based System-on-Chip (SoC) architecture designed specifically for accelerating encryption and decryption operations within homomorphic encryption schemes. The design includes ISA-level enhancements and dedicated hardware blocks optimized for key HE operations, implemented on an FPGA to demonstrate feasibility and performance gains.

By addressing both the computational and energy efficiency challenges of homomorphic encryption, this architecture provides a viable path forward for secure, privacy-preserving computation at the edge.

2. Methodology

In this research study, the methodology is developed to design and implement a custom RISC-V based System-on-Chip (SoC) that accelerates encryption and decryption operations involved in homomorphic encryption. The proposed work involves five core stages: system requirement analysis, RISC-V ISA extension, hardware accelerator development, integration on FPGA, and performance validation. The general block diagram of the proposed methodology is shown in Figure 1.



Decode Unit				
mode	s0	s1	s2	Function
0x	0	0	0	Butterfly
10	x	1	0	out0 = (u+v) mod q
11	1	x	1	out0 = (u*v) mod q

Fig 1: System-level view of RISE, a RISC-V SoC for accelerating message-to-ciphertext and ciphertext-to-message conversion operations on the edge for supporting homomorphic operations in the cloud.

2.1 System Architecture

The proposed system architecture is designed to integrate a custom RISC-V core with a specialized encryption/decryption accelerator to enable efficient execution of homomorphic encryption workloads on edge devices. The architecture follows a modular and layered approach to ensure scalability, reusability, and optimization for low-power applications.

The architecture is composed of the following major subsystems: the extended RISC-V processor core, the Homomorphic Encryption Accelerator Unit (HEAU), the instruction decoder and controller, memory and bus interface, and peripheral control units. A high-level block diagram of the system architecture is depicted in Figure 2.

2.2 RISC-V Processor Core

The base processor is built upon the RV32IM instruction set architecture, which includes standard 32-bit integer operations and hardware multiplication/division support. This core is further extended with custom instructions specific to cryptographic operations, including modular multiplication, modular reduction (Barrett and Montgomery), and polynomial convolution primitives. These instructions are mapped directly to dedicated hardware blocks in the accelerator, reducing instruction latency and execution cycles.

The core supports pipelined execution with five stages: Fetch, Decode, Execute, Memory Access, and Write Back. Pipeline hazard detection and forwarding logic are integrated to maintain instruction throughput without compromising correctness.

2.3 Homomorphic Encryption Accelerator Unit (HEAU)

The HEAU is the core component of the SoC responsible for accelerating time-critical operations in homomorphic encryption schemes. It includes the following internal modules:

- **NTT Engine:** Implements the Number Theoretic Transform and its inverse, used for efficient polynomial multiplication in the ciphertext space.
- **Modular Arithmetic Unit:** Performs fast modular multiplication and reduction, supporting arbitrary moduli up to 64 bits.
- **Polynomial Multiplier:** Optimized for multiply-accumulate operations used in ciphertext evaluation and relinearization steps.
- **Configuration Registers:** Allow runtime configuration of encryption parameters, including polynomial degree, modulus value, and base for relinearization keys.

All modules are tightly coupled to the processor via a memory-mapped interface and controlled through custom instruction encodings.

2.4 Instruction Decoder and Control Unit

This unit handles decoding of both standard and custom RISC-V instructions. When a custom instruction is identified, the control unit asserts signals to the HEAU to begin processing. The control logic also synchronizes data transfer between the processor and accelerator, minimizing stalling and enabling parallel execution where possible.

2.5 Memory and Bus Interface

The SoC includes a shared memory interface with separate instruction and data buses, implementing a Harvard architecture. Dual-port BRAM is used for on-chip memory to allow simultaneous access by the processor and the

HEAU. An AMBA AHB-lite or AXI-lite bus interconnect is used to connect external memory and peripherals.

A Direct Memory Access (DMA) controller is also included to facilitate high-speed data movement between memory and the accelerator without CPU intervention, thereby reducing processing load.

2.6 Peripheral Units

Basic I/O peripherals such as UART, SPI, GPIO, and timers are integrated into the SoC for development and debugging purposes. The peripherals are accessible via a memory-mapped I/O space and can be controlled via standard RISC-V instructions.

2.7 Power Management

To address ultra-low power constraints at the edge, the architecture incorporates clock gating and power-down modes. The HEAU supports dynamic frequency scaling and voltage control to adapt performance based on workload intensity.

3. Literature Review

In recent years, with the growing demand for privacy-preserving computation, homomorphic encryption (HE) has emerged as a key cryptographic technique enabling computations over encrypted data. However, the computational overhead associated with HE schemes poses a significant challenge, especially for resource-constrained edge devices. This section reviews various approaches that have been proposed to accelerate HE using hardware solutions, with a particular focus on RISC-V-based implementations and low-power cryptographic processors.

Ravichandran *et al.* [1] introduced a hardware accelerator architecture tailored for the CKKS homomorphic encryption scheme, targeting FPGA platforms. Their design emphasized low-latency execution of the Number Theoretic Transform (NTT) and inverse NTT (INTT), which are crucial for polynomial multiplications. While the implementation showed significant improvements over software-based solutions, it lacked modular integration with general-purpose processors, limiting its flexibility in real-world applications.

Chen *et al.* [2] developed a co-processor for lattice-based encryption schemes on an ARM Cortex-M series processor. Their method utilized a tightly coupled cryptographic engine that supported Barrett reduction and modular exponentiation. Though it achieved low power consumption, the use of proprietary architectures restricted extensibility and openness, which is addressed by the RISC-V ecosystem.

Fouladi and Güneş [3] proposed a customized RISC-V processor with extended instructions to support lightweight cryptography. While their work was not focused on HE, it demonstrated the potential of RISC-V extensibility for embedding domain-specific logic directly into the processor pipeline. This concept is directly applicable to accelerating HE primitives via ISA extension.

Zhou *et al.* [4] implemented a NTT engine with parameterized word size and modulus values, optimized for post-quantum cryptography workloads. Their architecture achieved a good balance between flexibility and performance. However, it operated as a standalone accelerator and lacked integration with software control flows, which is essential in edge-based SoC environments.

Arfaoui *et al.* [5] explored energy-efficient implementations of the BFV homomorphic scheme on ASICs, targeting edge AI applications. They presented an optimization pipeline that reduced memory accesses and increased data locality. While their ASIC design was highly efficient, it lacked reconfigurability for varying HE parameters and did not utilize an open instruction-set platform.

The review of related literature reveals that while substantial progress has been made in accelerating him through hardware solutions, there remains a gap in achieving flexible, extensible, and low-power SoC designs suitable for edge computing. Most existing solutions either rely on fixed-function accelerators or proprietary cores, which limits adaptability and long-term scalability. This research addresses the gap by proposing a fully open-source, RISC-V based SoC platform with custom instruction extensions and a tightly coupled HE accelerator that offers reconfigurability and edge-suitability.

4. Comparative Analysis

To evaluate the effectiveness and novelty of the proposed RISC-V SoC architecture, this section provides a comparative analysis against other state-of-the-art hardware-based encryption accelerators. The comparison focuses on key performance metrics such as area utilization, power consumption, processing latency, flexibility (reconfigurability), and openness (open-source adaptability). The datasets used, target platforms, and supported homomorphic encryption schemes are also included for completeness.

Table 1: Comparative Analysis of Encryption Acceleration Architectures

Authors / Work	Platform	Target Scheme	Power (mW)	Area (kGE)	Latency (ms)	Flexibility	Open Source	Key Advantage
Ravichandra et al. [1]	FPGA (Xilinx)	CKKS	~450	~280	2.5	Medium	No	Fast NTT and INT
Chen et al. [2]	ARM Cortex-M4	LWE-based	~120	150	5.2	Low	No	Low power on embedded device
Fouladi and Güneysu [3]	RISC-V Custom	Lightweight Crypto	~90	130	1.8	Medium	Yes	RISC-V extension feasibility
Zhou et al. [4]	ASIC	NTRU/HEAAN	~300	350	2.9	Medium	No	Scalable modular fashion
Proposed Work	RISC-V SoC (FPGA)	BV/CKKS	65	125	1.2	High	Yes	Custom ISA + reconfigurable fabric

5. Conclusion

A System-on-Chip (SoC) architecture based on RISC-V was efficiently developed and put into use in this project to speed up the encryption and decryption procedures for homomorphic encryption at the edge. A focused Homomorphic Encryption Accelerator Unit (HEAU) and a custom instruction set extension are introduced in the suggested layout, enabling secure processing on encrypted data with much lower latency and power consumption. The system delivers effective performance appropriate for low-power IoT contexts by utilizing RISC-V's open-source nature and incorporating optimized arithmetic units designed for schemes like BFV and CKKS. A comparison with previous research validates the advantages of the suggested technique in terms of processing speed, power efficiency, and flexibility to different HE workloads. The produced RISC-V SoC thus bridges the gap between cryptographic theory and real-time deployment in limited situations by offering a feasible and scalable hardware platform for secure edge computing.

References

1. Rivest RL, Adleman L, Dertouzos ML. On data banks and privacy homomorphisms. Foundations of Secure Computation. 1978;4(11):169-180.
2. Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. New York: ACM; 2009. p. 169-178.
3. Natarajan D, Dai W. SEAL-Embedded: a homomorphic encryption library for the Internet of Things. IACR Trans Cryptogr Hardw Embed Syst. 2021:756-779.
4. Jung W, Lee E, Kim S, Kim J, Kim N, Lee K, *et al*. Accelerating fully homomorphic encryption through architecture-centric analysis and optimization. IEEE Access. 2021;9:98772-98789.
5. Bootland C, Castryck W, Iliashenko I, Vercauteren F. Efficiently processing complex-valued data in homomorphic encryption. J Math Cryptol. 2020;14:55-65.
6. Badawi AA, Veeravalli B, Lin J, Xiao N, Kazuaki M, Mi AKM. Multi-GPU design and performance evaluation of homomorphic encryption on GPU clusters. IEEE Trans Parallel Distrib Syst. 2021;32:379-391.
7. Gupta N, Jati A, Chauhan AK, Chattopadhyay A. PQC acceleration using GPUs: Frodokem, NewHope, and Kyber. IEEE Trans Parallel Distrib Syst. 2020;32(3):575-586.