INL/CON-25-83444-Revision-0

Employing a Hardware-in-the-Loop Approach to Realize a Fully Homomorphic Controller for a Small Modular Advanced High Temperature Reactor

June 2025

Robert s Lois II, Dane Sabo, Patrick Murphy, Luis Felipe Benitez, Dan Cole

Changing the World's Energy Future

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/CON-25-83444-Revision-0

Employing a Hardware-in-the-Loop Approach to Realize a Fully Homomorphic Controller for a Small Modular Advanced High Temperature Reactor

Robert s Lois II, Dane Sabo, Patrick Murphy, Luis Felipe Benitez, Dan Cole

June 2025

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Employing a Hardware-in-the-Loop Approach to Realize a Fully Homomorphic Encrypted Controller for a Small Modular Advanced High Temperature Reactor

Robert S. Lois[1]*, **Dane Sabo**[1], **Patrick Murphy**[1], **Luis Felipe Benitez**[1], **Daniel G. Cole**[1]

[1]University of Pittsburgh, Pittsburgh, Pennsylvania

[leave space for DOI, which will be inserted by ANS]

ABSTRACT

This paper addresses the cybersecurity challenges of advanced nuclear reactors by integrating fully homomorphic encryption (FHE) into their control systems, enabling encrypted processing of control signals without compromising functionality.

Advanced nuclear reactors, including Small Modular Reactors (SMRs) and microreactors, aim to achieve autonomous and remote operations, reducing costs and enhancing competitiveness. However, these advancements expand the attack surface for cyberattacks, particularly in autonomous and remote operation scenarios. Cyberattacks can exploit vulnerabilities to manipulate physical processes, causing shutdowns, asset damage, or public harm. Such attacks begin with passive reconnaissance, where adversaries intercept communications or observe behaviors to gather information, which is then leveraged to execute cyber-physical attacks by injecting malicious commands. Nuclear power must adopt cybersecurity protection measures to secure the integrity and availability of their digital control systems.

This paper demonstrates the application of FHE to secure operations by enabling encrypted processing of sensitive signals and parameters – ensuring privacy without exposing data. FHE supports secure mathematical operations on encrypted data without requiring decryption. Using a hardware-in-the-loop (HIL) approach, this paper implements an FHE-integrated controller on a BeagleBone Black (BBB) controlling a simulation of the Small Modular Advanced High Temperature Reactor (SmAHTR). By doing so, the encrypted controller protects the integrity of critical set points and control signals during transmission and processing. Thus, FHE-integrated controllers enhance secure operations of advanced nuclear reactors while maintaining functionality.

Keywords: nuclear instrumentation and control systems, advanced reactors, secure operations, cybersecurity, cryptography

1. INTRODUCTION

1.1. Fully Autonomous and Remote Operations of Advanced Nuclear Reactors

Advanced nuclear reactors, including small modular reactors (SMRs) and microreactors, aim to overcome limitations of conventional nuclear systems with enhanced safety, scalability, and reduced environmental impact [1]. Despite these benefits, economic challenges such as high capital costs and regulatory barriers remain significant [2]. To address these issues, innovations like modular construction, digital twins, and advanced automated operations are being employed to reduce costs and improve efficiency [3].

*robert.lois@pitt.edu

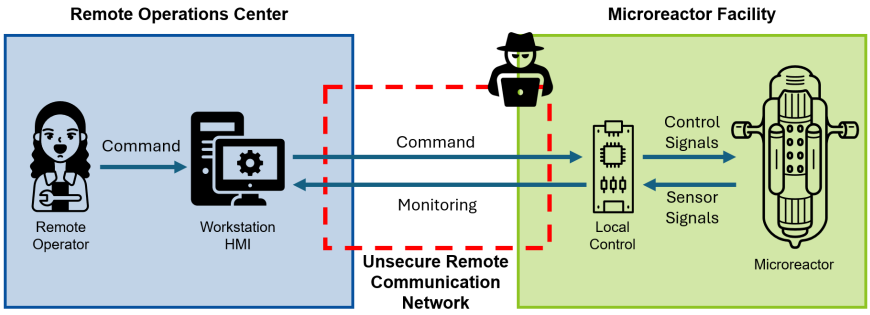


Figure 1: High-level reference architecture for remote operations of a microreactor where the remote operations center is communicating to the microreactor facility over an unsecure network.

Automation plays a crucial role in economic optimization, ranging from human-assisted systems to fully autonomous operations [4]. Current reactors rely on lower-level automation, where systems support human operators by monitoring and assisting with routine tasks [5]. In contrast, fully autonomous systems manage operations independently, enabling remote control from centralized centers. This approach reduces staffing needs, minimizes onsite infrastructure, and addresses workforce challenges in remote deployments [6].

Fully autonomous and remote operations pose technical and security challenges. Advanced instrumentation and control (I&C) systems must manage complex tasks, such as autonomous power adjustments, while supporting secure remote monitoring and control [7]. Robust communication infrastructure and cybersecurity measures are essential to prevent cyber-physical attacks, including malicious command injections and man-in-the-middle exploits, which could compromise safety and functionality [8].

Privacy-enhancing technologies, such as Fully Homomorphic Encryption (FHE), are key to securing data in autonomous systems. FHE allows encrypted computations, protecting sensitive data during transmission and processing without sacrificing functionality [9]. By integrating FHE into the digital I&C infrastructure, advanced reactors can achieve secure autonomous and remote operations, maintaining both privacy and functional compliance.

This paper demonstrates the integration of an FHE cryptosystem into a controller on a BeagleBone Black (BBB) microcomputer for the Small Modular Advanced High Temperature Reactor (SmAHTR). Using hardware-in-the-loop simulation, the encrypted control system controls rod position to regulate coolant temperature to given set points. Efficient memory use mitigates computational overhead, demonstrating the feasibility of FHE-based controllers for secure autonomous and remote reactor operations.

1.2. Architecture for Remote Operations of Microreactors

This paper aims to enhance the security of communication networks used in semi-to-fully autonomous operations with remote monitoring and control capabilities. We consider the architecture described by Culler et al. [10] and illustrate an abstracted version shown in Figure 1. The remote operations center is where reactor operators have monitoring and control capabilities, but is geographically separated from the microreactor facility. Within the microreactor facility, we assume autonomous operations with local control loops that maintain operation within their safety limits. As a result, a networked communication infrastructure is required for a remote operator to monitor and control the microreactor. The network infrastructure contains bidirectional communication between the remote operations center and the microreactor facility. The remote control room houses components such as operator workstations and human-machine-interfaces (HMIs). The microreactor facility has its own localized control systems, where feedback control loops manage control signals sent to the microreactor and process sensor signals observed from it. Commands from the remote operations center, such as set-point commands, are sent through the communication network to the microreactor facility. Data sent from the microreactor facility to the remote operator provides real-time monitoring of system performance and operational status.

1.3. Problem Statement and Threat Model

We consider a scenario where a remote operator transmits control commands and receives sensor feedback over an unencrypted network between the remote operations center and a microreactor's local control loop. In this setup, a passive adversary can intercept and analyze plaintext communications, extracting sensitive information about operational parameters, system dynamics, and control strategies as illustrated in Figure 1. The absence of encryption exposes the system to significant risks, including future targeted attacks or system manipulation. Thus, the primary security goal is to ensure end-to-end confidentiality of bidirectional communication, protecting set-point commands, control parameters, sensor measurements, and actuator signals from adversarial inference.

2. LEARNING WITH ERRORS-BASED CRYPTOSYSTEM

The goal of this research is to integrate a fully homomorphic encryption (FHE) scheme into microreactor control systems. We construct a learning with errors (LWE)-based public-key encryption (PKE) scheme that supports homomorphic operations, including Gentry-Sahai-Waters (GSW) ciphertext-based multiplication. The LWE PKE system is defined on the LWE assumption [11] and is parameterized by the secret key length N, lattice dimension n, ciphertext modulus q, and error distribution χ.

Definition 1 (Syntax) *A homomorphic PKE system based on* $\mathrm{LWE}_{N,n,q,\chi}$ *is defined by the following functions* $\varphi = (\mathrm{KeyGen}, \mathrm{LWE}, \mathrm{LWE}^{-1}, \mathrm{Eval})$*:*

Key Generation: $\mathrm{KeyGen}(1^\lambda) \to (sk, pk)$. *For some security parameter,* λ*, defined by* $\mathrm{LWE}_{N,n,q,\chi}$*, generate secret key* $sk : s \leftarrow \mathbb{Z}_q^N$ *and public key* $pk : (A, \nu = As + e)$ *where* $A \leftarrow \mathbb{Z}_q^{n\times N}$ *and* $e \sim \chi^n$*.*
Encryption: $\mathrm{LWE}(pk, m) \to c$*. This function takes an input the public key,* pk*, and a message,* $m \in \mathbb{Z}_t \subset \mathbb{Z}_q$*, generates a random vector* $r \leftarrow \{-1, 0, 1\}^n$*, and produces the ciphertext* $c = (c_1, c_2) \in \mathbb{Z}_q^{N+1}$*:*

$$\mathrm{LWE}\,(pk, m) := (c_1, c_2) = \left(A^\intercal r, \nu^\intercal r + \left\lfloor \frac{q}{t} \right\rfloor m\right) \tag{1}$$

Decryption: $\mathrm{LWE}^{-1}(sk, c) \to m$*. This function takes an input the secret key,* sk*, and a ciphertext,* $c \in \mathbb{Z}_q$*, and extracts the message* $m \in \mathbb{Z}_t$*.*

$$\mathrm{LWE}^{-1}(sk, c) := \left\lceil \frac{t}{q}\left(c_2 - s^\intercal c_1\right)\right\rfloor = \left\lceil \frac{t}{q}\left(\nu^\intercal r + \left\lfloor \frac{q}{t} \right\rfloor m - s^\intercal A^\intercal r\right)\right\rfloor \tag{2}$$

Evaluation: $\mathrm{Eval}(pk, f, \vec{c}) \to c_f$*. This function takes inputs the public key,* pk*, and a function* f *to perform homomorphic operations on ciphertexts,* $\vec{c} = (c_1, \ldots, c_\ell)$*, and outputs a ciphertext,* c_f*, that is an encryption of* $f(m_1, \ldots, m_\ell)$*.*

2.1. Gadget Decomposition and the Gentry-Sahai-Waters Ciphertext

The Gentry-Sahai-Waters (GSW) encryption scheme was proposed by Gentry, Sahai, and Waters [12] and is based on the LWE problem. This scheme introduces the concept of the gadget decomposition transformation. This transformation achieves homomorphic multiplication that corresponded directly to matrix multiplication while preserving correctness [13], which is necessary for our research in achieving encrypted control.

First, we present the gadget matrix as a generalized definition followed by its inverse transformation property. Let $F = \eta\ell$, where $\ell = \lfloor \log_b q \rfloor$ for a chosen base b, $\eta \geq 1$ as a positive integer and let $g = [1, b, b^2, \cdots, b^{\ell-1}] \in \mathbb{Z}_q^\ell$. The gadget matrix, $G \in \mathbb{Z}_q^{F\times\eta}$, is defined as:

$$G = \mathbb{I} \otimes g = \begin{pmatrix} 1 & \cdots & b^{\ell-1} & & & \\ & 1 & \cdots & b^{\ell-1} & & \\ & & \ddots & & \ddots & \\ & & & 1 & \cdots & b^{\ell-1} \end{pmatrix} \tag{3}$$

where $\mathbb{I}$ is an $\eta \times \eta$ identity matrix, $\otimes$ is the Kronecker product, and all unspecified entries are assumed to be zero. We now define the decomposition transformation function and its inverse property of the gadget. The decomposition transformation is a function $G^{-1} : \mathbb{Z}_q^\eta \to \mathbb{Z}_q^F$ such that for any vector $\beta \in \mathbb{Z}_q^\eta$:

$$G^{-1}(\beta) = [\beta_1, \beta_2, \ldots, \beta_\eta] \tag{4}$$

where each β_i is reconstructed from its radix-b representation $\beta_i = \sum_{j=0}^{\ell-1} \beta_{i,j} \cdot b^j$ with $\beta_{i,j}$ being the j-th digit of β_i in base b and $\beta_{i,j} \in \{0, 1, \ldots, b-1\}$. The inverse transformation property ensures the reconstruction of any vector β such that:

$$G^{-1}(\beta)G = \beta \in \mathbb{Z}_q^\eta. \tag{5}$$

Using the gadget transformation from Equation 3, the GSW encryption function is defined as:

Definition 2 (GSW Encryption) *Given* $\mathrm{LWE}_{N,n,q,\chi}$*, the encryption function,* $\mathrm{GSW}(pk, m) \to \mathbb{C}$*, takes as an input the public key,* pk*, and a message,* $m \in \mathbb{Z}_t \subset \mathbb{Z}_q$*, and produces the ciphertext* $\mathbb{C} \in \mathbb{Z}_q^{(N+1)F \times (N+1)}$*:*

$$\mathrm{GSW}\,(pk, m) = mG + \mathbb{O}$$

where $F = \lfloor \log_b q \rfloor$ *for some base b,* G *is the gadget, and* $\mathbb{O}$ *is a matrix of size* $\mathbb{Z}_q^{(N+1)F \times (N+1)}$ *where each row is an* LWE *encryption of zero such that* $\mathrm{LWE}\,(pk, 0)$*.*

2.2. Homomorphic Operations

In this section, we define the homomorphic operations for addition and multiplication. We demonstrate that addition is a straightforward operation between LWE ciphertexts. For multiplication, we introduce the external product between LWE and GSW ciphertexts that is derived from the Chillotti-Gama-Georgieva-Izabachene (CGGI) scheme [14].

Definition 3 (Homomorphic Addition) *Let* $c = \mathrm{LWE}(pk, m_1)$ *and* $c' = \mathrm{LWE}(pk, m_2)$ *be two correctly formed ciphertexts of messages* $m_1, m_2 \in \mathbb{Z}_t \subset \mathbb{Z}_q$*. Homomorphic addition between these ciphertexts produces the ciphertext* $c_\oplus \in \mathbb{Z}_q^{N+1}$*:*

$$c \oplus c' \to \mathrm{LWE}(pk, m_1 + m_2) = \left(A_1^\intercal r + A_2^\intercal r, \nu_1^\intercal r + \nu_2^\intercal r + \left\lfloor \frac{q}{t} \right\rfloor (m_1 + m_2)\right) \tag{6}$$

where $\oplus$ *denotes homomorphic addition and* $\to$ *denotes that the result of the homomorphic addition is an* LWE *ciphertext where the messages* m_1 *and* m_2 *were added.*

For homomorphic multiplication, we use the external product defined in the CGGI scheme [14] between an LWE and GSW ciphertext. The scheme demonstrated that the external product is more efficient than the previous work of homomorphically multiplying GSW ciphertexts. As such, we define homomorphic multiplication as follows:

Definition 4 (Homomorphic Multiplication) *Let* $c = \mathrm{LWE}(pk, m_1)$ *and* $\mathbb{C} = \mathrm{GSW}(pk, m_2)$ *be two correctly formed ciphertexts of messages* $m_1, m_2 \in \mathbb{Z}_t \subset \mathbb{Z}_q$*. Homomorphic multiplication between these ciphertexts produces the ciphertext* $\mathrm{LWE}(pk, m_1 m_2) \in \mathbb{Z}_q^{N+1}$*:*

$$c \odot \mathbb{C} \to \mathrm{LWE}(pk, m_1 m_2) = G^{-1}(c) \cdot \mathbb{C} \tag{7}$$

where $\odot$ *denotes homomorphic multiplication,* $G^{-1}(c)$ *decomposes the* $c = \mathrm{LWE}(pk, m_1)$ *ciphertext, and the* $\to$ *denotes that the result of the homomorphic multiplication as a* LWE *ciphertext for the product between* m_1 *and* m_2*.*

3. EXPERIMENTAL SETUP AND METHODS

The experimental setup for this research includes a BeagleBone Black (BBB) to serve as a controller-in-the-loop that interacts with the Advanced Reactor Cyber Analysis and Development Environment (ARCADE) and the Small Advanced High Temperature Reactor (SmAHTR). SmAHTR is a Fluoride-Salt-Cooled microreactor, designed to be highly portable with a target of $125\,\mathrm{MWt}$ [15]. This research uses a Simulink model of SmAHTR, which captures the behavior of four (4) individual reactor cores, one large salt vault, and a power generating secondary system using a Brayton cycle. For more information about the Simulink model and its capabilities please refer to [16,17].

The SmAHTR Simulink model consists of several proportional-integral controllers for flow-rate, mass-fraction, and reactivity controls. For this experiment, we replace the reactivity controller for Reactor 1 with an external proportional controller on the BBB. The controller receives measurements of the outlet coolant temperature of Reactor 1, calculates an error from a desired outlet coolant temperature reference value, and then applies a proportional gain as an actuator command. This command emulates control rod manipulation as a means of controlling reactivity for Reactor 1, resulting in a change in outlet coolant temperature. All other control systems in the Simulink model were unchanged.

ARCADE is an open-source hardware-in-the-loop platform that is built to use precompiled or Simulink simulations and communicate with control systems using the MODBUS communication protocol. ARCADE consists of two main components: a data broker and a collection of endpoints. The data broker communicates with the SmAHTR simulation through shared memory, manages the execution of simulation timesteps, and distributes sensor signals to the endpoints. Endpoints are the interfaces that connect ARCADE and SmAHTR system components with external systems. This enables the exchange of data and control signals necessary for reactor monitoring, control, and analysis. For more information about ARCADE and its capabilities, please refer to [18].

The hardware used as the controller is the BeagleBone Black (BBB) microcomputer. It features a 1 GHz ARM Cortex-A8 processor, 512 MB of DDR3L RAM, and 4 GB of onboard eMMC flash storage, with a Debian-based Linux operating system. Using Python and the PyModbus package, the controller was implemented in two forms: (1) a plaintext, unencrypted version and (2) an encrypted, learning with errors (LWE)-based version. The plaintext controller was realized using standard Python libraries.

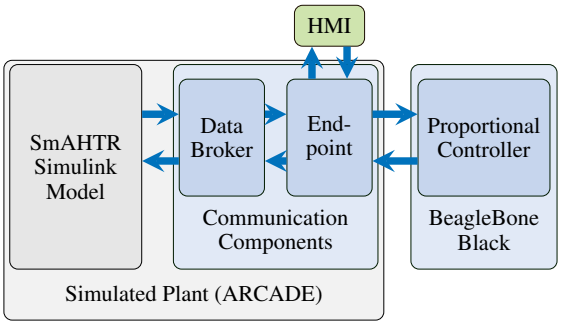
For realizing the encrypted controller, several modifications were required. First, custom-built Python modules of the LWE-based cryptosystem, as defined in Section 2, were created. Additionally, a MODBUS packet handler was developed for transmitting ciphertexts, as detailed in Section 4.2. Finally, the ARCADE platform was extended to include an interface for encryption and decryption between the SmAHTR model and the encrypted controller. This interface serves as a system for key generation, encrypting control parameters, sensor measurements, and reference setpoints, and decrypting actuator commands. This process is further discussed in Section 4.

In addition to the LWE-based interface, the endpoint was modified to include a human-machine interface (HMI), which serves as a model of remote operations of the SmAHTR reactor. The HMI allows the operator to set a reference signal and monitor system performance. The experimental setup for the plaintext controller is depicted in Figure 2a and the encrypted controller with its modifications are depicted in Figure 2b.

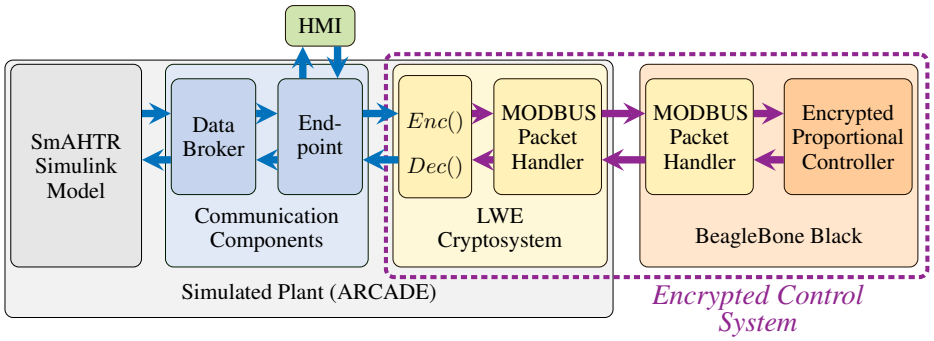
4. REALIZATION OF ENCRYPTED CONTROLLER-IN-THE-LOOP

In this section, we implement an encrypted controller for a hardware-in-the-loop experiment. The framework described in [19] is used to realize an encrypted proportional controller with the LWE-based cryptosystem described in Section 2 on the BeagleBone Black (BBB). Using discrete-time state space representation, the encrypted time-invariant proportional control loop is define by:

$$\text{End-to-End Encrypted Control Loop} \quad \begin{cases} \hat{y}_k = \mathrm{LWE}(pk, Cx_k^p) \\ \hat{u}_k = G^{-1}(\hat{r}_k - \hat{y}_k)\hat{K}_p \\ u_k = \mathrm{LWE}^{-1}(sk, \hat{u}_k) \end{cases} \tag{8}$$



(a) The plaintext controller-in-the-loop experimental setup.



(b) The encrypted controller-in-the-loop experimental setup, including LWE cryptosystem functions and the MODBUS packet handler.

Figure 2: Experimental setup of ARCADE platform for the plaintext and encrypted controller-in-the-loop that interacts with the SmAHTR Simukink model.

where $x_k^p \in \mathbb{R}^\eta$ denotes the state vector of the physical system at time step k and $C \in \mathbb{R}^{m\times\eta}$ is the output matrix. Using Equation 1, the sensor measurement, $y_k = Cx_k^p$ and the reference input, r_k, at time step, k, are encrypted as $\hat{y}_k = \text{LWE}(pk, y_k)$ and $\hat{r}_k = \text{LWE}(pk, r_k)$, respectively. Using Equation 2, the proportional gain, K_p, is encrypted as $\hat{K_p} = \text{GSW}(pk, K_p)$. The encrypted actuator command, at time step k, is computed through homomorphic operations using Equation 6 and Equation 7 as $\hat{u}_k = G^{-1}(\hat{r}_k - \hat{y}_k)\hat{K_p}$. Prior to system state update, the actuator command is decrypted using Equation 2 as $u_k = \text{LWE}^{-1}(sk, \hat{u}_k)$. This formulation ensures that all computations, from sensor measurement through the controller and to the actuator, are performed on encrypted data, thus preserving confidentiality throughout the entire control loop.

4.1. Initialization of Encrypted Proportional Controller

Prior to system startup, the learning with errors (LWE)-based public-key encryption (PKE) scheme is instantiated as $\text{LWE}_{N,n,q,\chi}$ within the endpoint of the ARCARDE platform. The instantiation is parameterized by positive integers N, n, and q, which define the secret key length, the lattice dimension, and the ciphertext modulo space, respectively, and additionally with the error distribution χ. For χ, this research uses the discrete Gaussian distribution [20], which is centered at zero with a width defined by σ. The selection of the remaining parameters follow the framework from [19]. The public key, $pk = (A, \nu) \in \mathbb{Z}_q^{n\times N} \times \mathbb{Z}_q^n$, and the secret key, $sk = s \in \mathbb{Z}_q^N$ are then generated, thereby establishing the LWE-based cryptosystem for secure encrypted controller operations within the endpoint interface.

Using the public key, the proportional gain is encrypted, $\hat{K_p} = \text{GSW}(pk, K_p)$, and transmitted to the BBB-controller via MODBUS TCP/IP. Upon reception, the encrypted control loop is initialized by setting the encrypted reference value, $\hat{r}_k = \text{LWE}(pk, r_k)$, supplied by the human-machine-interface (HMI). Once

the reference is set, the BBB-controller makes encrypted observations through the endpoint interface as $\hat{y}_k = \text{LWE}(pk, Cx_k^p)$. Within the BBB-controller, the encrypted proportional control law is performed as $\hat{u}_k = G^{-1}(\hat{r}_k - \hat{y}_k)\hat{K}_p$. The encrypted actuator command is sent to the endpoint interface where it is decrypted, $u_k = \text{LWE}^{-1}(sk, \hat{u}_k)$, and interacts with the SmAHTR simulink model.

4.2. Sending Encrypted Messages with MODBUS

The Pymodbus Python library is used to facilitate communications between the BBB and ARCADE using MODBUS TCP/IP. MODBUS TCP/IP is a messaging service based on MODBUS, which is an application message protocol frequently used in industrial automation and is available on industry standard PLCs. MODBUS TCP/IP encapsulates the MODBUS message and adds a header to make it usable across TCP connections. Within the MODBUS protocol there is a maximum standard Protocol Data Unit (PDU) length of 253 bytes with one byte reserved for the function code, leaving 252 bytes for data [21]. These remaining bytes are mapped to a maximum of 126 registers per frame for 2 bytes of PDU data per register [22].

Due to the large size of LWE-based ciphertexts [23], transmitting them over MODBUS requires careful fragmentation and reassembly of data into packet segments since it can occupy at most 252 bytes (126 registers). Consider an arbitrary, correctly formed ciphertext, $C \in \mathbb{Z}_q^{\alpha\times\beta}$ and let $c_{ij} \in \mathbb{Z}_q$, for $1 \leq i \leq \alpha$ and $1 \leq j \leq \beta$, denote the elements of C. This implies a total ciphertext size of C as $\mathcal{O}(\alpha\beta\log(q))$ bits with its elements as $\mathcal{O}(\log(q))$ bits. Consequently, each ciphertext, $C \in \mathbb{Z}_q^{\alpha\times\beta}$ must satisfy $\alpha\beta\lceil\log(q)\rceil \leq 252\,\text{bytes} \cdot 8\,\text{bits}$, to ensure that it can be transmitted in a single MODBUS frame. Thus, we developed a MODBUS packet handler that satisfies this condition and splits the ciphertext across multiple frames prior to transmission and reconstructed upon reception.

4.3. Theoretical Model for Control Loop Latency

This research defines the control loop latency by the round-trip time, denoted as T_{RTT}, which represents the total time required for a control cycle, from sensor measurement to actuator response. It accounts for all delays introduced by the MODBUS communication and computation within our controller-in-the-loop setup. Formally, the control loop latency is expressed as:

$$T_{\text{RTT}} = T_{\text{write}} + T_{\text{wait}} + T_{\text{read}} \tag{9}$$

where T_{write} is the time required to a write a value to a MODBUS register, including network transmission and the encryption process if applicable, T_{wait} is the computational delay for controller execution, and T_{read} is the time required to read a value from a MODBUS register, including network transmission and the decryption process if applicable.

Unique to the ARCADE platform is a predefined scan time within the endpoint to synchronize control loop operations. This scan time is directly analogous to T_{wait}, representing a *waiting* period for the controller to complete its computations before performing a reading operation. Because of this implementation, the scan time is known *a priori* and is set to $T_{\text{wait}} = 100\,\text{ms}$ for this research.

5. RESULTS

The learning with errors (LWE)-based public-key encryption (PKE) system was instantiated with the secret key length and lattice dimension $N = n = 5$, the ciphertext modulo space $q = 2^{32}$, and discrete Gaussian width set to $\sigma = 5$. The proportional gain for the encrypted and plaintext controller was set to $K_p = 0.005$. The encrypted controller was initialized as described in Section 4.1. After system start up, the reactor coolant temperature reaches equilibrium with an initial reference coolant temperature of $T_0 = 640\,^\circ\text{C}$. A step change in the reference signal is introduced to raise the desired reactor outlet coolant temperature to $T_{ref} = 650^\circ$.

The response in the reactor coolant temperature to the step change for both controllers is shown in Figure 3. From the results, we can visually evaluate the performance of the encrypted control system by comparing it to the plaintext controller. The system responses for these controllers are similar in shape and steady-state performance. This implies that the LWE-based cryptosystem does not degrade the performance of

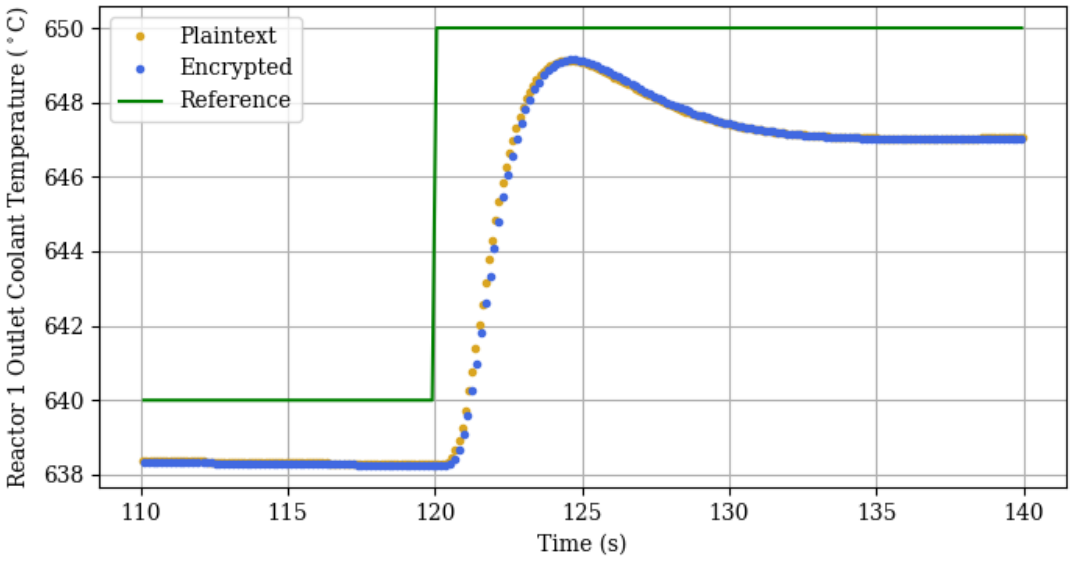


Figure 3: Simulated results of reactor outlet coolant temperature control with an encrypted and plaintext controller-in-the-loop.

the controller during runtime. Additionally, the total round-trip time recorded for the plaintext controller was $T^{pt}_{\text{RTT}} = 115\,\text{ms}$, while the encrypted controller was measured as $T^{\text{enc}}_{\text{RTT}} = 151\,\text{ms}$, reflecting an expected increase in latency. The delay in $T^{\text{enc}}_{\text{RTT}}$ is due to the increased $T^{\text{enc}}_{\text{read}}$ and $T^{\text{enc}}_{\text{write}}$, which is caused by the overhead of writing and reading the larger ciphertexts to and from the MODBUS registers. From the recorded data, each plaintext signal and proportional gain has a size of $4\,\text{bytes}$, whereas the corresponding LWE signal expands to $48\,\text{bytes}$ and the GSW proportional gain expands to $3456\,\text{bytes}$ for the encrypted controller. This significant increase in message size contributes to the additional latency observed in the encrypted control loop.

6. CONCLUSIONS & FUTURE WORK

This work successfully demonstrates integrating a fully homomorphic encryption scheme (FHE) into advanced nuclear reactor control systems, achieving the primary security goal and addressing the challenges in autonomous and remote operations. Using a hardware-in-the-loop (HIL) simulation with a BeagleBone Black (BBB) microcontroller managing a Small Modular Advanced High Temperature Reactor (SmAHTR) model, we successfully implemented a learning with errors (LWE)-based proportional controller capable of securely processing encrypted control. Despite computational overhead in the encrypted control loop, real-time performance was maintained, highlighting the practicality of LWE-based control for safeguarding sensitive operational data without compromising functionality.

Despite the success of the encrypted controller-in-the-loop, certain aspects of the experimental setup warrant further discussion regarding potential areas for improvement. When considering the results in Figure 3, neither controller tracks the input reference signal. This deviation comes from replacing a proportional-integral controller with a proportional controller. Without an integral term, the controller lacks the ability to compensate for state-steady errors, leading to the offset from the desired reference in our results. As such, future work is to realize an encrypted proportional-integral controller-in-the-loop and analyze its performance.

Additional future work includes analyzing the computational complexity of the encrypted control loop when varying the LWE parameters, this includes increasing the secret key length and its overall impact on controller performance. We also recognize the limitations of LWE-based schemes in terms of efficiency and operating on nonlinear functions. Future efforts will explore integrating advanced schemes that employ ring learning with errors (RLWE) in order to enhance performance and support nonlinear control systems.

NOMENCLATURE

This paper uses notation to suit a control engineering reader and may deviate from common notation used in cryptographic research. As such, we denote $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$ as the set of natural numbers, integers, and real numbers, respectively. The set of integers modulo q for $q \in \mathbb{Z}$ is denoted by the finite field $\mathbb{Z}_q$ of cardinality q and its elements in the range $(-q/2, q/2]$ and $[x]_q$ is the reduction of x into $\mathbb{Z}_q$ such that $[x]_q = x \bmod q$. The round, floor, and ceiling function are denoted by $\lceil \cdot \rfloor$, $\lfloor \cdot \rfloor$, and $\lceil \cdot \rceil$, respectively. All vectors are column vectors by default unless noted otherwise. For $m \in \mathbb{N}$ and $n \in \mathbb{N}$, we denote the size of a matrix as $A^{m \times n}$. Vector multiplications are denoted by $a \cdot b$ and matrix multiplications are denoted without the 'dot'. For appending vectors and matrices, we use the notation (b, a) for vector b appended to the front of matrix a. The transpose of a vector or a matrix A is denoted by $A^\intercal$. For sampling from a distribution χ, we use the notation $x \sim \chi^n$ for x of size n. For uniform sampling from a set $\mathbb{S}$, we use the notation $x \leftarrow \mathbb{S}^n$ for x of size n. To denote a functions input-output relationship we use the $\rightarrow$ notation.

ACKNOWLEDGMENTS

This article was authored and/or co-authored by employees of Battelle Energy Alliance, LLC (BEA). BEA is the Management and Operating Contractor of the Idaho National Laboratory with the United States Department of Energy (DOE), under Contract Number DE-AC07-05ID14517 (BEA's DOE Prime Contract) with principal offices located at 2525 North Fremont Avenue, P.O. Box 1625, Idaho Falls, ID 83415-3899. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish and reproduce the published form of this article, or allow others to do so, for United States Government purposes.

REFERENCES

[1] T. G. Lane and S. T. Revankar. "Advances in technology, design and deployment of microreactors-a review." *Progress in Nuclear Energy*, **volume 178**, p. 105520 (2025).

[2] World Nuclear Association. "Economics of Nuclear Power." (n.d.). URL https://world-nuclear.org/information-library/economic-aspects/economics-of-nuclear-power. Accessed: 2024-12-27.

[3] H. C. Bryan, K. W. Jesse, C. A. Miller, and J. M. Browning. "Remote nuclear microreactors: a preliminary economic evaluation of digital twins and centralized offsite control." *Frontiers in Nuclear Engineering*, **volume 2**, p. 1293908 (2023).

[4] A. L. Alberti, V. Agarwal, I. Gutowska, C. J. Palmer, and C. R. de Oliveira. "Automation levels for nuclear reactor operations: A revised perspective." *Progress in Nuclear Energy*, **volume 157**, p. 104559 (2023).

[5] U.S. Nuclear Regulatory Commission (NRC). "NUREG-0700, Revision 3: Human-System Interface Design Review Guidelines." (n.d.). URL https://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0700/r3/index.html. Accessed: 2024-12-27.

[6] E. S. Fleming Lindsley, M. Nyre-Yu, and D. L. Luxat. "Human Factors Considerations for Automating Microreactors." Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States) (2020).

[7] K. Stevens and et al. "Opportunities, Challenges, and Research Needs for Remote Microreactor Operations." *Nuclear Technology*, pp. 1–17 (2024).

[8] B. Zohuri. "Enhancing Nuclear Reactor Safety through ICE Digital Systems and Cyber Integration." *Journal of Material Sciences & Manufacturing Research SRC/JMSMR-198 DOI: doi org/1047363/JMSMR/2024 (5)*, **volume 164**, pp. 2–6 (2024).

[9] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. Fitzek, and N. Aaraj. "Survey on fully homomorphic encryption, theory, and applications." *Proceedings of the IEEE*, **volume 110**(10), pp. 1572–1609 (2022).

[10] M. Culler, J. Oncken, K. Stevens, and T. Ulrich. "Architecture Design for Remote Operation of Microreactors." In *2024 Resilience Week (RWS)*, pp. 1–10. IEEE (2024).

[11] O. Regev. "On lattices, learning with errors, random linear codes, and cryptography." *Journal of the ACM (JACM)*, **volume 56**(6), pp. 1–40 (2009).

[12] C. Gentry, A. Sahai, and B. Waters. "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based." In *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pp. 75–92. Springer (2013).

[13] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. "TFHE: fast fully homomorphic encryption over the torus." *Journal of Cryptology*, **volume 33**(1), pp. 34–91 (2020).

[14] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachene. "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds." In *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22*, pp. 3–33. Springer (2016).

[15] S. R. Greene and et al. "Pre-Conceptual Design of a Fluoride-Salt-Cooled Small Modular Advanced High Temperature Reactor (SmAHTR)." Technical Report ORNL/TM-2010/199, Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States) (2011).

[16] C. Dangelo and D. Cole. "Hot Standby State Observers for Sensor Fault-Tolerance in Small Modular Reactors." In *American Nuclear Society Winter Meeting* (2015).

[17] J. A. Farber and D. G. Cole. "Real-Time Supervisory Control Implementation of SmAHTR Power Plant." *10th Int Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies*, pp. 322–330 (2017).

[18] A. S. Hahn. "Sandialabs/SMARTT: Small Modular Reactor Testing and Training." https://github.com/sandialabs/SMARTT.

[19] R. S. Lois and D. G. Cole. "Encrypted Control Using Modified Learning with Errors-based Schemes." *IFAC-PapersOnLine*, **volume 58**(28), pp. 72–77 (2024).

[20] N. C. Dwarakanath and S. D. Galbraith. "Sampling from discrete Gaussians for lattice-based cryptography on a constrained device." *Applicable Algebra in Engineering, Communication and Computing*, **volume 25**, pp. 159–180 (2014).

[21] "MODBUS Application Protocol Specification V1.1b3." (2012). URL https://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

[22] G. Thomas. "Introduction to the modbus protocol." *The Extension*, **volume 9**(4), pp. 1–4 (2008).

[23] H. Nejatollahi, N. Dutt, and R. Cammarota. "Special session: Trends, challenges and needs for lattice-based cryptography implementations." In *2017 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*, pp. 1–3. IEEE (2017).