# Adaptive Lightweight Encryption-Based Secure UART Architecture for IoT Devices

**Ajas Ahamed F[1], Mohamed Rafeek K[2], Nihal F[3], Ms.Menaga[4]**
[1,2,3,4] Department of Electronics and Communication Engineering
[1,2,3] Mohamed Sathak A.J College of Engineering
[4] Assiatant Professor, Mohamed Sathak A.J College of Engineering

***Abstract-*** *The rapid growth of IoT applications requires secure, energy-efficient, and reliable communication for resource-constrained devices. UART is widely used in IoT systems due to its simplicity and low hardware cost, but conventional secure UART designs typically employ a single lightweight cryptographic algorithm, limiting adaptability to changing power, security, and performance requirements.*

*This paper proposes an Adaptive Lightweight Encryption-Based Secure UART Architecture that dynamically selects among three lightweight block ciphers—GIFT, PRESENT, and LBlock-S—based on real-time system conditions. A Cipher Selection Controller monitors power level, security needs, and speed requirements to activate the most suitable algorithm: GIFT for low-power operation, PRESENT for higher security, and LBlock-S for high-speed communication.*

*The design is implemented in Verilog HDL and validated using Xilinx Vivado FPGA simulations. Results demonstrate accurate cipher selection, seamless encryption switching, and reliable UART communication without data loss. The proposed architecture provides a flexible, hardware-efficient, and context-aware secure communication solution for IoT devices.*

***Keywords:*** IoT Security, Secure UART, Lightweight Cryptography, Adaptive Encryption, FPGA Implementation, Embedded Systems

## I. INTRODUCTION

### A. Growth of IoT and Security Challenges

The Internet of Things (IoT) has emerged as a transformative paradigm enabling billions of interconnected devices to sense, process, and exchange data across diverse application domains, including smart cities, industrial automation, healthcare monitoring, and consumer electronics. According to recent industry reports, the number of connected IoT devices is expected to exceed 25 billion in the coming years, highlighting the scale and impact of this technology [1]. Despite its rapid growth, IoT ecosystems face significant challenges related to security, privacy, and reliability due to the inherently constrained nature of edge devices.

IoT nodes are typically limited in terms of processing capability, memory resources, and energy availability. These constraints make the direct adoption of conventional cryptographic algorithms, such as Advanced Encryption Standard (AES) or RSA, impractical for many low-power embedded platforms [2]. As a result, IoT systems are often deployed with minimal or inadequate security mechanisms, exposing them to threats such as eavesdropping, data manipulation, replay attacks, and unauthorized access. Secure communication remains one of the most critical requirements in IoT architectures, particularly for applications involving sensitive data such as medical records or industrial control signals.

Furthermore, IoT environments are highly dynamic, with varying operational conditions including fluctuating power levels, changing security requirements, and diverse performance demands. Static security solutions that do not account for these variations may lead to inefficient resource utilization or weakened protection. Consequently, there is a growing demand for adaptable and lightweight security mechanisms that can operate efficiently under diverse IoT scenarios [3].

### *B. Importance of UART in Embedded Systems*

Universal Asynchronous Receiver Transmitter (UART) is one of the most widely used serial communication protocols in embedded and IoT systems due to its simplicity, low cost, and minimal hardware requirements. UART facilitates point-to-point asynchronous communication between microcontrollers, sensors, and peripheral devices without the need for complex clock synchronization mechanisms [4]. As a result, UART is extensively employed in IoT sensor nodes, debugging interfaces, bootloaders, medical devices, and industrial embedded platforms.

Despite its widespread adoption, standard UART communication does not provide inherent security features such as encryption or authentication. Data transmitted over UART is typically sent in plain text, making it vulnerable to interception and manipulation, especially in exposed or hostile environments [5]. While several secure UART implementations have been proposed, most existing solutions integrate a single cryptographic algorithm into the data path, resulting in a fixed-security configuration.

This fixed-cipher approach limits flexibility and fails to address the diverse operational conditions encountered in real-world IoT deployments. For example, a device operating on battery power may prioritize energy efficiency over high security, whereas the same device may require stronger

encryption when transmitting sensitive data. Therefore, enhancing UART communication with adaptive security mechanisms is essential for achieving robust and efficient IoT communication.

### *C. Overview of Lightweight Cryptography*

To address the limitations of conventional cryptography in constrained environments, lightweight cryptographic algorithms have been developed specifically for low-power and low-area applications. Lightweight cryptography focuses on reducing hardware complexity, energy consumption, and latency while maintaining an acceptable level of security [6]. Several lightweight block ciphers have been standardized or widely adopted for IoT use cases.

**PRESENT** is a well-known lightweight block cipher that employs a substitution–permutation network (SPN) structure and offers strong security with minimal hardware overhead [7]. It has been extensively studied and is suitable for applications requiring higher security guarantees.

**GIFT** is a more recent lightweight block cipher designed with hardware efficiency and resistance to side-channel attacks in mind, making it particularly suitable for ultra-low-power IoT devices [8].

**LBlock** and its variants, such as LBlock-S, are optimized for high-speed encryption and low latency, making them suitable for performance-critical embedded systems [9].

While these lightweight ciphers are individually effective, most existing secure communication designs rely on a single cipher, selected at design time. This static selection does not exploit the complementary strengths of different ciphers and fails to adapt to changing system requirements.

### *D. Motivation for Adaptive Encryption*

The heterogeneous and dynamic nature of IoT applications motivates the need for adaptive security mechanisms capable of responding to real-time operational constraints. A fixed lightweight cipher may perform well under certain conditions but may be suboptimal under others. For instance, a cipher optimized for low power may not provide sufficient security for sensitive data transmission, while a high-security cipher may consume excessive energy when used continuously.

Adaptive encryption addresses this limitation by dynamically selecting the most appropriate cryptographic algorithm based on contextual parameters such as power availability, security requirements, and data rate. By enabling real-time cipher selection, an adaptive system can balance security, performance, and energy efficiency more effectively than static approaches [10]. Despite its potential benefits, adaptive encryption has received limited attention in the context of UART-based communication for IoT devices.

Integrating adaptive encryption into UART communication introduces several challenges, including synchronization between transmitter and receiver, hardware complexity, and real-time decision-making. However, advances in FPGA technology and modular hardware design make it feasible to implement such adaptive architectures efficiently.

### *E. Contributions of This Work*

Motivated by the limitations of fixed-cipher secure UART designs and the need for flexible IoT security solutions, this paper makes the following key contributions:

- **An adaptive secure UART architecture** that dynamically selects among multiple lightweight block ciphers based on real-time operational conditions.
- **A Cipher Selection Controller** that monitors power level, security requirement, and speed demand to autonomously determine the optimal encryption algorithm.
- **Integration of three lightweight ciphers—GIFT, PRESENT, and LBlock-S**—leveraging their complementary strengths in power efficiency, security, and performance.
- **A hardware implementation using Verilog HDL**, validated through simulation on the Xilinx Vivado FPGA platform.
- **Comprehensive simulation results** demonstrating correct cipher switching, secure data transmission, and stable UART operation without data loss.

This paper proposes an adaptive lightweight encryption-based secure UART architecture that enhances the flexibility, efficiency, and security of UART communication in resource-constrained IoT devices.

## II. BACKGROUND AND RELATED WORK

### *F. Lightweight Cryptography for IoT Systems*

Lightweight cryptography has emerged as a critical research area to address the stringent resource constraints of IoT and embedded devices. Unlike conventional cryptographic algorithms such as AES and RSA, which require substantial computational power and memory, lightweight cryptographic primitives are specifically designed to minimize hardware area, energy consumption, and latency while maintaining adequate security levels [11]. These characteristics make lightweight ciphers suitable for applications such as wireless sensor networks, RFID systems, and low-power IoT nodes. Among lightweight block ciphers, **PRESENT** is one of the earliest and most widely studied designs. It follows a substitution–permutation network (SPN) structure and supports 80-bit and 128-bit keys with a 64-bit block size. Due to its compact hardware implementation and strong cryptographic properties, PRESENT has been adopted in various embedded security applications [7]. However, its relatively higher number of rounds may result in increased latency and energy consumption when used continuously. **GIFT** is a more recent lightweight block cipher designed with hardware efficiency and resistance to side-channel attacks as primary objectives. It improves upon earlier designs by offering a bit-sliced architecture that is particularly efficient in hardware implementations [8]. GIFT has been recognized for its ultra-low-power operation, making it suitable for energy-

constrained IoT environments. Nevertheless, GIFT prioritizes power efficiency over high throughput, which may limit its applicability in performance-sensitive systems. **LBlock** and its optimized variants, such as **LBlock-S**, are lightweight Feistel-based block ciphers designed to achieve high encryption speed with low latency [9]. These ciphers are well-suited for applications requiring fast data processing but may not always provide the same level of energy efficiency as ultra-lightweight designs like GIFT. The existence of multiple lightweight ciphers with complementary strengths highlights the potential benefits of adaptive cipher selection in dynamic IoT environments.

### *G. Secure Communication in IoT and Embedded Systems*

Secure communication is a fundamental requirement in IoT architectures to ensure data confidentiality, integrity, and authenticity. Numerous studies have investigated security mechanisms for IoT communication protocols, including MQTT, CoAP, and 6LoWPAN [3], [5]. While these protocols provide security features at higher layers, low-level communication interfaces within embedded systems often remain unprotected. UART-based communication is widely used for inter-device data exchange, debugging, firmware updates, and sensor interfacing. Despite its prevalence, UART lacks built-in security mechanisms, making it vulnerable to physical and logical attacks such as data sniffing and injection [4]. Several researchers have proposed hardware-based secure communication frameworks to protect serial data paths in embedded systems. Some works integrate symmetric encryption modules directly into UART transmitters and receivers to encrypt outgoing data and decrypt incoming data [12]. These approaches improve confidentiality but typically employ a single fixed cryptographic algorithm, resulting in static security behavior. While effective under certain conditions, fixed-cipher designs fail to adapt to changing operational requirements such as power constraints or varying security levels.

Other studies have explored FPGA-based security solutions for embedded communication, leveraging reconfigurable hardware to implement cryptographic primitives efficiently [13]. These designs demonstrate the feasibility of hardware-level security but often focus on performance or area optimization rather than adaptability. As a result, the majority of existing secure UART implementations remain inflexible and application-specific.

### *H. Adaptive and Context-Aware Security Approaches*

Adaptive security mechanisms aim to dynamically adjust security parameters in response to environmental or system-level changes. Context-aware cryptographic systems have been explored in software-defined networks, mobile computing, and cloud environments to balance security and performance [10]. In the context of IoT, adaptive security is particularly relevant due to the heterogeneous and dynamic nature of device operation.

Some research efforts have proposed adaptive cryptographic frameworks that adjust key sizes, encryption modes, or protocol parameters based on resource availability [14]. These approaches demonstrate that adaptability can significantly improve energy efficiency and system lifetime. However, most existing adaptive schemes are implemented at higher protocol layers or in software, making them less suitable for low-level embedded communication interfaces.

Hardware-based adaptive encryption remains relatively underexplored, especially for UART communication. Implementing adaptability at the hardware level introduces challenges related to synchronization, control logic complexity, and verification. Nevertheless, advances in FPGA technology and modular hardware design have made it possible to integrate multiple cryptographic cores and selection logic with manageable overhead.

To the best of the authors' knowledge, very limited work has addressed **adaptive lightweight encryption specifically for UART-based communication** in IoT devices. Existing designs either focus on a single lightweight cipher or lack real-time adaptability, leaving a significant research gap in this domain.

### *I. Research Gap and Motivation*

From the above discussion, it is evident that while lightweight cryptography and secure UART designs have been extensively studied, most existing solutions suffer from one or more of the following limitations:

1. **Use of a single fixed lightweight cipher**, leading to suboptimal performance under varying conditions.
2. **Lack of context awareness**, with no consideration of power level, security sensitivity, or throughput requirements.
3. **Limited hardware adaptability**, particularly at the UART interface level.

Given the dynamic operational nature of IoT devices, a secure communication architecture that can adapt its cryptographic behaviour in real time is highly desirable. This observation motivates the development of an adaptive secure UART architecture that leverages the complementary strengths of multiple lightweight ciphers.

By integrating cipher selection logic directly into the hardware design, the proposed approach aims to provide a flexible, efficient, and scalable security solution for IoT embedded systems. This work addresses the identified research gap by introducing an FPGA-based adaptive lightweight encryption mechanism tailored specifically for UART communication.

## II. Proposed Adaptive Secure UART Architecture

This section presents the detailed architecture and operation of the proposed **Adaptive Lightweight Encryption-Based Secure UART**. The objective of the proposed design is to provide a flexible, context-aware, and hardware-efficient secure communication mechanism for IoT and embedded devices by dynamically selecting the most suitable lightweight cipher based on real-time operational conditions. The architecture integrates multiple lightweight cryptographic

cores with a Cipher Selection Controller and a secure UART communication interface.

### *A. System Overview*

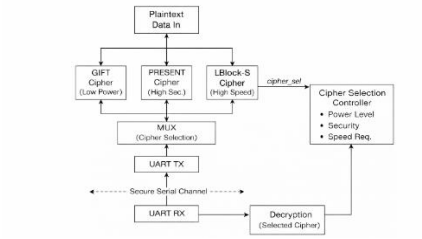


*Figure 1 Architecture of the proposed adaptive secure UART*

The overall architecture of the proposed adaptive secure UART system is illustrated in **Figure 1**. The design consists of four main functional blocks: the **Cipher Selection Controller**, **Lightweight Cipher Modules**, **Encryption Selection Multiplexer**, and the **Secure UART Transmitter and Receiver**. These components work together to enable adaptive encryption and secure serial communication.

At the transmitter side, the plaintext data to be transmitted is provided as input to the system. This data is simultaneously fed into multiple lightweight cipher modules operating in parallel. The Cipher Selection Controller continuously monitors system-level parameters such as power level, security requirement, and speed demand. Based on these inputs, it generates a cipher selection signal that determines which encryption output is forwarded for transmission.

The encrypted data selected by the multiplexer is then passed to the UART transmitter module, which handles serial data transmission according to UART protocol principles. On the receiver side, the incoming encrypted data is received through the UART receiver and decrypted using the same cipher mode selected at the transmitter, ensuring data correctness and synchronization.

The parallel instantiation of cipher modules combined with real-time selection logic enables seamless switching between encryption algorithms without requiring system reconfiguration or communication interruption. This architectural approach significantly enhances flexibility and adaptability compared to conventional fixed-cipher secure UART designs.

**Data Flow Description:**

1. Plaintext data is applied to the encryption subsystem.
2. All cipher modules encrypt the data simultaneously.
3. The Cipher Selection Controller evaluates system conditions.
4. The multiplexer selects the appropriate encrypted output.
5. The selected ciphertext is transmitted via the UART interface.
6. The receiver performs synchronized decryption using the same cipher mode.

This modular and scalable architecture allows additional ciphers or selection parameters to be incorporated in future extensions with minimal redesign effort.

### *B. Cipher Selection Controller*

The **Cipher Selection Controller** is the core decision-making unit of the proposed adaptive secure UART architecture. Its primary function is to dynamically select the most appropriate lightweight cipher based on real-time operational constraints. This controller enables context-aware security by balancing power efficiency, security strength, and communication performance.

**Inputs to the Controller**

The controller operates based on the following input parameters:

- **Power Level Indicator:** Represents the available energy or operating mode of the device (e.g., low, medium, or high power).
- **Security Requirement Level:** Indicates the sensitivity of the data being transmitted and the required level of cryptographic strength.
- **Speed Requirement Signal:** Specifies whether high data throughput or low latency communication is required.

These inputs can be derived from system monitors, battery management units, application-level configurations, or external control logic in a real-world deployment.

**Decision Logic**

The Cipher Selection Controller employs a combinational decision logic that maps the input parameters to a cipher selection signal. The logic prioritizes constraints based on predefined rules that reflect typical IoT operating scenarios. For instance, when power availability is critically low, energy-efficient encryption is prioritized over high security or throughput. Conversely, when transmitting sensitive data, stronger encryption is selected even if it incurs higher computational cost.

The controller generates a multi-bit selection signal that controls the encryption multiplexer. This signal determines which cipher output is forwarded to the UART transmitter. The decision process is lightweight and incurs negligible hardware overhead, making it suitable for real-time operation in constrained environments.

**Decision Cases**

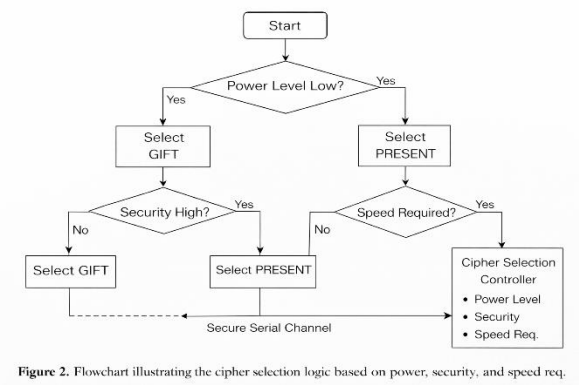


*Figure 2 chipper selection flow chart*

The main cipher selection cases implemented in the proposed system are summarized as follows and illustrated in **Figure 2**:

- **Ultra-Low-Power Mode:** When the power level is low, the controller selects the **GIFT cipher**, which is optimized for minimal energy consumption and hardware efficiency.
- **High-Security Mode:** When the security requirement is high, the controller selects the **PRESENT cipher**, which provides stronger cryptographic security through its substitution–permutation structure.
- **High-Speed Mode:** When high throughput or low latency is required, the controller selects the **L Block-S cipher**, which is optimized for fast encryption and reduced delay.
- **Default Mode:** In the absence of strict constraints, a default cipher is selected to ensure stable operation.

This rule-based decision mechanism ensures predictable behaviour while enabling dynamic adaptability. The controller can be easily extended to include additional parameters or more complex decision strategies if required.

### *C. Cipher Modules*

The proposed architecture integrates three lightweight block ciphers: **GIFT**, **PRESENT**, and **L Block-S**. Each cipher is selected based on its distinct performance characteristics, enabling the system to adapt to diverse operational requirements.

**Design Philosophy**

The cipher modules are designed with the following principles in mind:

- **Lightweight Hardware Implementation:** Each cipher is implemented using simple combinational and sequential logic to minimize area and power consumption.
- **Modularity:** Each cipher module is implemented as an independent hardware block with standardized input and output interfaces, enabling easy integration and scalability.
- **Complementary Strengths:** The selected ciphers collectively cover a wide range of performance metrics, including power efficiency, security strength, and throughput.

Rather than relying on a single cryptographic primitive, the proposed system leverages the strengths of multiple ciphers to provide a balanced and adaptable security solution.

**Parallel Instantiation**

All cipher modules are instantiated **in parallel**, and the plaintext input is broadcast to each cipher simultaneously. This parallelism eliminates the need for reconfiguration delays when switching between ciphers and ensures immediate availability of encrypted data for selection.

The outputs of the cipher modules are connected to a multiplexer controlled by the Cipher Selection Controller. Only one encrypted output is selected at any given time, while the others remain inactive. Although parallel instantiation slightly increases hardware usage, the overhead is justified by the significant improvement in adaptability and responsiveness.

This approach also simplifies control logic and avoids complex state transitions, making the design robust and easier to verify.

### *D. Secure UART Operation*

The secure UART subsystem integrates encryption seamlessly into the serial communication process without altering the fundamental UART protocol structure. This ensures compatibility with existing UART-based systems while enhancing security.

**Encryption Process**

Before transmission, plaintext data is encrypted using the selected lightweight cipher. The encryption process occurs entirely in hardware, providing faster execution and improved security compared to software-based approaches. The encrypted data maintains the same data width as the original input, enabling direct integration with the UART transmitter.

Because the cipher selection is determined prior to transmission, the encryption process does not introduce additional latency beyond the cipher computation itself.

**UART Transmission**

The UART transmitter serializes the encrypted data and transmits it over the communication channel using standard UART signalling conventions. The secure UART design

preserves the simplicity and low overhead of UART while adding a cryptographic protection layer.

The transmitter remains idle when no data is available and initiates transmission only when enabled by the control logic. This behaviour ensures efficient utilization of communication resources.

**Synchronization and Decryption**

To ensure correct data recovery, the receiver must use the same cipher mode selected at the transmitter. In the proposed architecture, synchronization is achieved through shared system configuration or predefined control signalling. The UART receiver deserializes the incoming encrypted data and applies the corresponding decryption operation using the selected cipher.

This synchronized encryption–decryption process guarantees data integrity and correctness without requiring complex key exchange or protocol modifications. The design assumes a secure key provisioning mechanism, which can be implemented as part of future enhancements.

## III. HARDWARE IMPLEMENTATION ON FPGA

This section describes the hardware realization of the proposed adaptive lightweight encryption-based secure UART architecture. The complete design is implemented using **Verilog Hardware Description Language (HDL)** and validated on an FPGA platform using the **Xilinx Vivado Design Suite**. The implementation emphasizes modularity, parallelism, and efficient hardware utilization to ensure suitability for resource-constrained IoT devices.

### *E. Verilog HDL Design Methodology*

The proposed secure UART architecture is modeled entirely in Verilog HDL, enabling precise control over hardware behavior and efficient synthesis on FPGA devices. A register-transfer level (RTL) design approach is adopted, where each functional component of the system is described as an independent Verilog module. This approach enhances design clarity, verification, and reusability.

The Verilog implementation includes modules for lightweight encryption, cipher selection logic, multiplexing, and UART transmission and reception. All modules are synthesized using standard IEEE-compliant Verilog constructs, ensuring portability across FPGA platforms. Combinational logic is primarily used for cipher selection and multiplexing, while sequential logic is employed in the UART transmitter and receiver for clocked data transfer.

### *F. Modular Architecture*

A key feature of the implementation is its **modular structure**, where each functional block is implemented as a separate Verilog module with well-defined interfaces. The main modules include:

- Cipher Selection Controller
- GIFT Cipher Module
- PRESENT Cipher Module
- L Block-S Cipher Module
- Encryption Output Multiplexer
- UART Transmitter (TX)
- UART Receiver (RX)

This modular organization allows individual components to be tested independently and facilitates future upgrades, such as adding new cipher modules or extending selection logic. It also simplifies debugging and improves overall design scalability.

### *G. Parallel Encryption Cores*

To enable real-time adaptive encryption without reconfiguration delays, the three lightweight cipher modules—GIFT, PRESENT, and LBlock-S—are instantiated **in parallel** within the top-level module. The plaintext input data is broadcast simultaneously to all encryption cores. Each cipher independently produces its encrypted output in the same clock cycle.

Parallel instantiation ensures that encrypted data from all ciphers is immediately available for selection, enabling seamless switching between encryption modes based on system conditions. Although this approach slightly increases hardware resource usage, it eliminates latency associated with sequential cipher execution and significantly improves adaptability. This trade-off is acceptable for FPGA-based IoT platforms where flexibility and responsiveness are prioritized.

### *H. MUX-Based Cipher Selection*

The encrypted outputs from the parallel cipher modules are connected to a **multiplexer**, which selects the appropriate ciphertext based on the control signal generated by the Cipher Selection Controller. The multiplexer is implemented using combinational logic, ensuring minimal selection delay and no impact on system throughput.

The selection signal (cipher_sel) is derived from real-time inputs representing power level, security requirement, and speed demand. Depending on the selected mode, the multiplexer forwards the corresponding encrypted data to the UART transmitter. This MUX-based approach provides a simple yet effective mechanism for dynamic cipher selection and avoids complex control state machines.

### *I. UART Transmitter and Receiver Integration*

The secure UART subsystem integrates encryption seamlessly into the data transmission path. The **UART transmitter** accepts encrypted data from the multiplexer and serializes it according to UART communication principles. The transmitter operates synchronously with the system clock and initiates data transmission upon receiving a transmission enable signal.

On the receiving side, the **UART receiver** deserializes incoming data from the secure serial channel. The decrypted data is obtained by applying the same cipher mode selected at

the transmitter. Synchronization between transmitter and receiver is ensured through consistent cipher selection logic, avoiding mismatches during encryption and decryption.

Importantly, the integration of encryption does not modify the fundamental UART protocol, preserving compatibility with existing UART-based systems while enhancing communication security.

### *J. Xilinx Vivado Design Flow*

The complete design is implemented and verified using the **Xilinx Vivado Design Suite**. The hardware development flow includes the following steps:

1. Creation of an RTL project and addition of Verilog source files
2. Functional simulation using Vivado's behavioral simulation environment
3. RTL elaboration and schematic generation
4. Design synthesis and logic optimization
5. Resource utilization analysis

Functional simulation confirms correct cipher selection, encrypted data routing, and UART transmission under different operational scenarios. The RTL schematic generated by Vivado, shown in **Figure 3**, visually verifies the correct interconnection of cipher modules, multiplexer, selection controller, and UART blocks. The schematic clearly demonstrates parallel encryption cores and MUX-based adaptive selection, validating the intended architecture.

### *K. RTL Schematic Analysis*

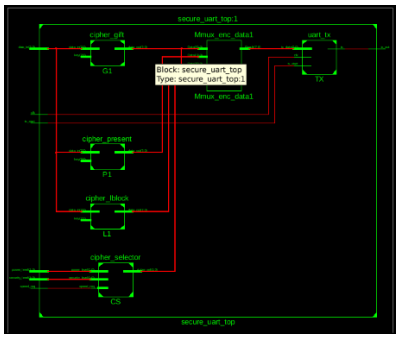


*Figure 3 RTL Schematic*

**Figure 3** presents the RTL schematic of the proposed adaptive secure UART architecture as generated by Vivado. The schematic highlights the parallel instantiation of the GIFT, PRESENT, and LBlock-S cipher modules, the central multiplexer for encrypted data selection, and the Cipher Selection Controller driving the selection logic. The UART transmitter and receiver blocks are shown integrated with the encryption subsystem, confirming the correctness of the overall hardware design.

The absence of unintended logic duplication or unconnected signals indicates a clean and optimized RTL implementation. The schematic further confirms that the design is synthesizable and suitable for FPGA deployment.

**Summary**

The FPGA-based hardware implementation demonstrates the feasibility and effectiveness of the proposed adaptive secure UART architecture. By leveraging Verilog HDL, modular design principles, parallel encryption cores, and MUX-based selection logic, the system achieves real-time adaptability with minimal overhead. The integration with UART communication and validation through Xilinx Vivado establish the design as a practical and scalable solution for secure IoT communication.

## IV. SIMULATION RESULTS AND ANALYSIS

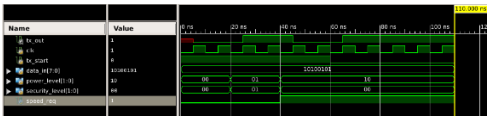


*Figure 4 Simulation Results*

This section presents the functional simulation results and performance analysis of the proposed adaptive lightweight encryption-based secure UART architecture. The objective of the simulation is to validate correct operation of adaptive cipher selection, encryption functionality, and secure UART transmission under different system conditions. All simulations are carried out using the **Xilinx Vivado Design Suite** at the register-transfer level (RTL).

### *L. Simulation Setup*

The proposed design is verified using a dedicated Verilog testbench developed to emulate realistic IoT operating conditions. The testbench generates clock signals, plaintext input data, and control signals corresponding to power level, security requirement, and speed demand. These control inputs dynamically influence the Cipher Selection Controller, enabling validation of real-time cipher switching behavior.

The system clock is configured with a uniform period to ensure synchronous operation of all modules. Multiple simulation scenarios are executed by varying the control inputs during runtime without resetting the system. This approach validates the ability of architecture to adapt dynamically while maintaining stable UART communication.

The primary signals monitored during simulation include:

- Plaintext input data
- Cipher selection control signals
- Encrypted data outputs
- UART transmission signal (tx_out)
- Clock and control inputs

Waveforms and RTL schematics generated by Vivado are used to analyze functional correctness and data flow.

### *M. Functional Verification of Cipher Selection*

One of the key objectives of the simulation is to verify the correctness of the Cipher Selection Controller. The controller is tested under multiple operating modes to ensure that the appropriate lightweight cipher is selected based on real-time system constraints.

**Low-Power Mode**

When the power level input is set to a low state, the controller selects the **GIFT cipher** regardless of the security or speed requirements. Simulation results confirm that the cipher selection signal correctly activates the GIFT encryption output, which is then forwarded to the UART transmitter. This behavior validates the controller's prioritization of energy efficiency under constrained power conditions.

**High-Security Mode**

When the security requirement signal is asserted, the controller selects the **PRESENT cipher**. Simulation waveforms show that the encrypted output from the PRESENT module is routed through the multiplexer, while other cipher outputs remain inactive. This confirms that the architecture correctly prioritizes cryptographic strength when data sensitivity is high.

**High-Speed Mode**

When the speed requirement signal is asserted and power constraints are relaxed, the controller selects the **LBlock-S cipher**. The simulation demonstrates immediate switching to the LBlock-S encrypted output without interruption to UART transmission, highlighting the effectiveness of the parallel encryption and MUX-based selection strategy.

These results collectively confirm that the Cipher Selection Controller operates correctly and deterministically under all tested conditions.

### *N. Verification of Parallel Encryption and MUX Operation*

The simulation results also validate the parallel operation of the lightweight cipher modules. Plaintext data is simultaneously processed by all three cipher cores—GIFT, PRESENT, and LBlock-S—during each encryption cycle. The encrypted outputs are continuously available at the inputs of the multiplexer.

The multiplexer, controlled by the cipher selection signal, forwards only the selected encrypted output to the UART transmitter. Simulation waveforms confirm that:

- No contention occurs at the multiplexer inputs
- The selected output changes correctly with the cipher selection signal
- No transient glitches or undefined states are observed during switching

This confirms that the MUX-based selection mechanism is stable and suitable for real-time adaptive encryption.

### *O. Secure UART Transmission Analysis*

The secure UART transmitter is validated by observing the tx_out signal under different encryption modes. Simulation results show that the UART transmitter remains idle when transmission is disabled and initiates data transfer only when the transmission enable signal is asserted.

When encrypted data is provided to the UART transmitter, the tx_out signal reflects the serialized encrypted output. Importantly, changes in cipher selection do not disrupt ongoing UART transmission, demonstrating seamless integration of encryption into the UART data path.

On the receiver side, the UART receiver correctly captures the incoming encrypted data. Synchronized decryption using the selected cipher ensures correct data reconstruction. The absence of mismatches between transmitted and received data confirms the correctness of the encryption–decryption process.

### *P. Waveform Analysis*

The simulation waveforms, as shown in the results section, provide clear evidence of correct system behavior. The following observations are noted:

- The cipher selection signal transitions cleanly between different states corresponding to GIFT, PRESENT, and LBlock-S.
- The encrypted data output changes immediately following cipher selection updates.
- The UART output signal remains stable and free from glitches.
- Control signals and data signals remain synchronized with the system clock.

These waveform characteristics confirm that the proposed design meets timing and functional requirements at the RTL level.

### *Q. RTL Schematic Verification*

The RTL schematic generated by Vivado, shown earlier in **Figure 3**, further validates the correctness of the hardware implementation. The schematic clearly illustrates:

- Parallel instantiation of the three cipher modules
- Centralized multiplexer for encrypted data selection
- Dedicated Cipher Selection Controller
- Integrated UART transmitter and receiver blocks

The schematic confirms that the synthesized design matches the intended architectural specification and contains no redundant or unintended logic. This structural verification complements the functional simulation results.

### *R. Result Discussion*

The simulation results demonstrate several important advantages of the proposed architecture:

1. **Real-Time Adaptability:**

   The system dynamically switches between encryption algorithms without requiring reconfiguration or reset, making it suitable for dynamic IoT environments.

2. **Hardware Efficiency:**

   The use of lightweight ciphers and simple selection logic ensures minimal overhead while providing enhanced security.

3. **Seamless UART Integration:**

   Encryption is transparently integrated into the UART data path without altering protocol behavior.

4. **Scalability:**
   The architecture can be extended to include additional ciphers or selection parameters with minimal design changes.

While the current simulation focuses on functional correctness, the results strongly indicate that the proposed design can achieve a favorable balance between security, performance, and energy efficiency.

**Summary**

The simulation results confirm that the proposed adaptive lightweight encryption-based secure UART architecture operates correctly under diverse operating conditions. Dynamic cipher selection, parallel encryption, stable multiplexer operation, and secure UART transmission are all validated through RTL simulation. These results demonstrate the feasibility and effectiveness of the proposed approach for secure and adaptable IoT communication.

## CONCLUSION AND FUTURE WORK

### *S. Conclusion*

This paper presented adaptive **lightweight encryption-based secure UART architecture** designed to address the security and efficiency challenges of IoT and resource-constrained embedded systems. Unlike conventional secure UART implementations that rely on a single fixed cryptographic algorithm, the proposed architecture dynamically selects among multiple lightweight block ciphers—GIFT, PRESENT, and LBlock-S—based on real-time operational constraints such as power availability, security requirements, and speed demand.

The proposed system integrates a dedicated Cipher Selection Controller, parallel encryption cores, and a multiplexer-based selection mechanism to enable seamless and real-time cipher switching. The architecture was implemented using Verilog HDL and validated through RTL simulation on the Xilinx Vivado FPGA platform. Simulation results confirmed correct cipher selection, stable UART transmission, and reliable encryption–decryption operation without communication interruption. The modular and scalable design demonstrates that adaptive encryption can be effectively realized at the hardware level with minimal overhead.

By providing context-aware security and flexibility, the proposed architecture significantly improves the adaptability of UART-based communication in IoT environments. The results highlight the potential of adaptive lightweight cryptography as a practical solution for balancing security, performance, and energy efficiency in embedded communication systems.

### *T. Future Work*

Several directions for future research can further enhance the proposed system. First, full cryptographic implementations of GIFT, PRESENT, and LBlock-S with standardized key sizes and round structures can be integrated to strengthen security guarantees. Second, quantitative evaluation of power consumption, area utilization, and throughput on real FPGA hardware can provide deeper insights into performance trade-offs.

Future work may also incorporate **dynamic key management and secure key exchange mechanisms** to improve system robustness. Additionally, implementing countermeasures against side-channel attacks and fault injection attacks would further enhance security. Extending the architecture to support additional lightweight cryptographic algorithms and exploring **ASIC implementation** for ultra-low-power applications represent promising research directions.

Overall, the proposed adaptive secure UART architecture lays a strong foundation for future research in flexible and efficient hardware-based security solutions for IoT systems

## REFERENCES

[1]. L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010.
[2]. A. Perrig et al., “Security in wireless sensor networks,” Communications of the ACM, vol. 47, no. 6, pp. 53–57, 2004.
[3]. M. Conti et al., “Internet of Things security and forensics: Challenges and opportunities,” Future Generation Computer Systems, vol. 78, pp. 544–546, 2018.
[4]. J. Axelson, Serial Port Complete, 2nd ed., Lakeview Research, 2007.
[5]. S. Raza et al., “Securing communication in 6LoWPAN with compressed IPsec,” IEEE DCOSS, 2011.
[6]. NIST, “Lightweight Cryptography Project,” 2019.
[7]. A. Bogdanov et al., “PRESENT: An ultra-lightweight block cipher,” CHES, 2007.
[8]. B. Beaulieu et al., “The GIFT lightweight block cipher,” CHES, 2017.
[9]. W. Wu and L. Zhang, “LBlock: A lightweight block cipher,” ACNS, 2011.

[10]. S. Tillich et al., “Dynamic cryptographic systems for embedded security,” IEEE Embedded Systems Letters, 2016.
[11]. K. McKay et al., “Report on Lightweight Cryptography,” NIST, 2017.
[12]. S. Chattopadhyay et al., “Hardware-based secure serial communication for embedded systems,” IEEE Embedded Systems Letters, 2015.
[13]. M. Feldhofer and C. Rechberger, “A case against currently used lightweight block ciphers,” ACNS, 2006.
[14]. Y. Zhou et al., “Adaptive security mechanisms for IoT based on resource awareness,” IEEE Internet of Things Journal, 2019.