International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Low-Power Secure UART Design Using Data Scrambling and Error Detection

LAKSHMI PRIYADHARSHINI P

Master of Engineering in VLSI DESIGN

Dr. Sivanthi Aditanar College of Engineering Tiruchendur, India

lakshmipriyadharshini.p@gmail.com

Abstract - Serial links in embedded and IoT systems face two problems: channel noise can silently flip bits, and plain-text transmission can be read by anyone monitoring the line. This paper presents a low-power secure Universal Asynchronous Receiver/Transmitter (UART) architecture, built as a transmitter chain and a receiver chain joined by a single serial channel, that recovers the original data and provides reliable data transfer. In the transmitter chain, 8-bit parallel data is first passed to a CRC generator, which appends the check bits. The result is scrambled by an XOR scrambler driven by an LFSR keystream loaded with a TX seed, and the UART transmitter then adds the start, parity and stop bits and sends the frame serially. In the receiver chain, the UART receiver detects the start bit, samples the frame and rebuilds the parallel word. A descrambler with an LFSR loaded with the matching RX seed recovers the original data, and a CRC checker recomputes the CRC and compares it with the received value to ensure reliable data transfer. The data output block delivers the 8-bit word together with an error flag, which is raised on any mismatch caused by channel noise or a seed error. The design uses only shift registers, XOR gates and finite-state logic, which gives low area and low power, and clock gating of the CRC, LFSR and UART blocks cuts dynamic power when the link is idle. It is suited to FPGA and ASIC implementation in sensor networks, IoT nodes and industrial control. LFSR scrambling provides lightweight obfuscation, not strong encryption.

Keywords - UART, CRC, LFSR, data scrambling, error detection, clock gating, low power.

I. INTRODUCTION

Reliable data exchange over a simple serial link is a basic requirement in embedded and digital systems. The UART is widely used because it needs only one line per direction, requires no shared clock and is easy to implement in hardware. However, a standard UART gives no protection against channel noise and sends data in plain form, so corrupted bits can go unnoticed and the content can be read by anyone monitoring the line.

Many links run in noisy environments such as factory floors, vehicles and sensor installations, where a single flipped bit can change a sensor reading or a control command. Serial links may also carry sensitive values such as calibration or configuration data. Strong ciphers like AES need considerable logic, memory and power, which small devices often cannot afford. A practical alternative is to combine inexpensive techniques: a CRC gives dependable error detection with minimal hardware, LFSR scrambling randomizes the bit pattern and prevents casual reading, and UART supplies a widely supported framing format.This work combines these three techniques in a transmitter chain and a receiver chain joined by a single serial channel. The objectives are (i) to design a CRC - 4 generator and checker for detecting transmission errors and improving the reliability of UART data communication. (ii) To implement LFSR-based XOR data scrambling and descrambling for secure data transmission and recovery of the original data at the receiver.

II. LITERATURE REVIEW

LFSR-based scrambling is common in storage. Buddhanoy et al. [1] extracted data-scrambling keys from commercial 2D and 3D NAND flash by writing complementary patterns; the recovered keys were periodic (16 to 64 bytes), which shows that fixed LFSR keys are predictable. Raghavendra et al. [2] proposed an LFSR frequency divider with state extension, multi-stage feedback selection and clock gating in a very small footprint (2 LUTs, 5 registers), but reported no power results. Kartik et al. [3] compared 8- to 32-bit LFSRs on an Artix-7: area and power scaled linearly (0.076 W to 0.088 W) and timing stayed constant, although only the Fibonacci form was implemented and LFSR linearity was noted as a security weakness.

For lightweight security, Deng et al. [4] used NLFSR obfuscation of PUF responses, resisting modeling, impersonation and replay attacks with 884 LUTs and 500 FFs, though security was argued analytically. Hatmojo et al. [5] secured UART firmware updates with ASCON-128, adding only 7.48 KB of Flash and 2.03 KB of SRAM and rejecting all tampering at the first chunk, but ASCON was slower than AES-GCM in software.

On the UART side, Swetha et al. [6] integrated CRC into the serial data path; CRC-16 detected over 99% of faulty frames versus 28% for parity and 63% for Hamming, with LUT use up only 3 to 7%, but it offers no security. Karthigaa et al. [7] built a UART with

configurable parity (34 LUTs for TX, 38 for RX), yet parity catches only odd-numbered errors. Dhanasekar et al. [8] added FIFOs, 16x oversampling and majority voting (under 1% data loss at 4.2 Mbps) without actual area or power figures. Rajavel et al. [9] switched baud rate (9600, 19200, 115200 bps) by FIFO occupancy, verified only in simulation. Existing work therefore treats error detection, scrambling and security separately, and few designs combine low area, error detection and obfuscation, which motivates the proposed design.

Table: Comparison Table For Existing Method

LIST OF EXISTING BASE PAPERS	STRENGTH	LIMITATIONS
Key extraction from LFSR-based NAND flash data scrambling	Shows scrambling keys (16 to 64 bytes) can be recovered on 2D/3D, SLC/MLC devices	Reveals that fixed, periodic LFSR keys are predictable
NLFSR obfuscation of PUF responses for authentication	Resists modeling, impersonation, and replay attacks, with 884 LUTs and 500 FFs	Security argued analytically, with no formal proof or attack experiments
UART with RX/TX FIFOs, 16x oversampling, and majority voting	Under 1% data loss at 4.2 Mbps and 96% noise immunity	No actual power, area, or timing figures; comparison tables look illustrative
ASCON-128 authenticated encryption for UART firmware updates	Low memory overhead (7.48 KB Flash, 2.03 KB SRAM); tampering rejected at the first chunk	Slower than AES-GCM and ChaCha20 in software; no side-channel or fault-injection evaluation
UART with configurable even/odd parity using FSMs	Compact, under 1% of an Artix-7 device	Parity catches only odd-numbered bit errors; nonstandard baud rate (240500 bps); text and table figures disagree
Comparison of 8 to 32-bit LFSRs on FPGA	Area and power scale linearly; timing stays constant	Claims a Fibonacci vs. Galois comparison but implements only Fibonacci
LFSR frequency divider with state extension, feedback selection, and clock gating	Very small footprint (2 LUTs, 5 registers) and less predictable output	Uses a 4-bit LFSR; no power results despite the clock-gating claim
UART with baud rate switching by FIFO occupancy, 32-bit data	Avoids FIFO overflow under burst traffic in simulation	Simulation only; inconsistent descriptions of the switching rule; no area, power, or timing data
CRC-16 generation and checking integrated into the UART data path	Over 99% error detection with no throughput loss	Detects errors but does not correct them; no security features

III. DESIGN DESCRIPTION

A. Proposed Architecture

The proposed architecture in Fig. 1 is divided into a transmitter chain (top row) and a receiver chain (bottom row), joined by a single serial channel (TX/RX line). The transmitter chain consists of the parallel data input, CRC generator, XOR scrambler with LFSR (TX seed) and UART TX. The receiver chain consists of the UART receiver, descrambler with LFSR (RX seed), CRC checker and data output with error flag.

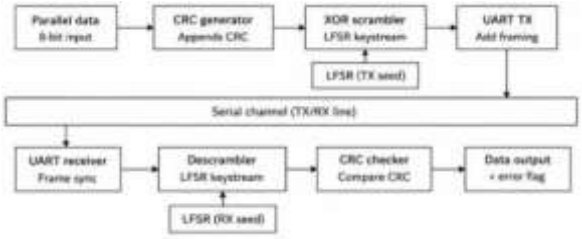


Fig 1: Block Diagram of the Proposed Low-Power Secure UART

B. Transmitter Side

The parallel data block holds the 8-bit word in a register of eight D flip-flops and presents it stably to the CRC generator. The CRC generator divides the data by a generator polynomial using XOR operations and appends the remainder. In this work a CRC-4 with G(x) = x^4 + x + 1 (binary 10011) is used, giving a 12-bit data-plus-CRC sequence. The XOR scrambler combines this sequence bit by bit with a keystream from a 4-bit LFSR loaded with a non-zero TX seed (for example 1101). The UART TX adds the start bit, optional parity bit and stop bit and sends the bits at the baud rate. Since the 12-bit sequence exceeds the 8-bit UART payload, it is sent across more than one frame.

C. Receiver Side

The UART receiver detects the start bit, samples the bits at the agreed baud rate, checks the stop bit, removes the framing and rebuilds the parallel word using an 8-bit serial-to-parallel shift register. The descrambler XORs the data with the keystream of a local LFSR having the same polynomial and a seed equal to the TX seed. Since $(A \oplus K) \oplus K = A$, the original data and CRC are restored. The CRC checker recomputes the CRC over the recovered data and compares it with the received value. The data output block delivers the 8-bit data with a valid signal and an error flag, which is raised on noise, a framing error or a seed mismatch.

D. Low-Power Control

Clock gating disables the clocks of the CRC, LFSR and UART blocks when the link is idle and enables them only during transmission and reception. This reduces dynamic power, and since the design contains only shift registers, XOR gates and simple control logic, the area is small.

IV. METHODOLOGY & IMPLEMENTATION

A. Methodology

The flowchart of the proposed design is shown in Fig. 2. A CRC is appended to the data, the data is scrambled with the LFSR keystream, framed by the UART and sent over the serial channel. The receiver deframes and synchronizes, descrambles with the same keystream and checks the CRC. If the CRC matches, the data is delivered to the application; otherwise an error is flagged and the frame is discarded or retransmitted. For this to work, both sides must use the same CRC polynomial and the same LFSR polynomial and seed; otherwise descrambling fails and the CRC check always reports an error.

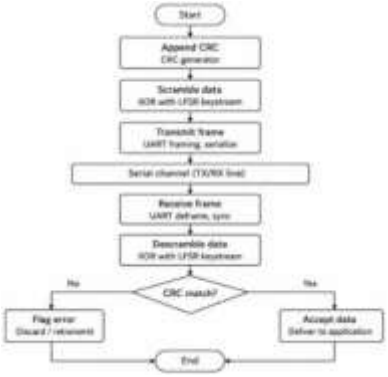


Fig 2: Flow-Chart of the Proposed Design

The algorithm follows these steps: (1) initialize and reset all blocks; (2) accept 8-bit parallel data; (3) append the CRC; (4) load the seed and generate the LFSR keystream; (5) XOR the data and CRC with the keystream; (6) add UART framing and transmit serially; (7) receive, detect the start bit, sample and rebuild the bits; (8) regenerate the keystream and descramble; (9) recompute the CRC and set CRC_MATCH or CRC_ERROR; (10) gate the clocks of inactive blocks; (11) output the data with the error flag and return to the input stage.

B. Implementation

The design is described in Verilog HDL and its blocks are modelled at gate level (D flip-flops and XOR logic) for simulation with Tanner S-Edit and T-Spice, followed by layout in L-Edit with design-rule and layout-versus-schematic checks. The existing UART with CRC-based error detection (baud-rate generator, UART TX/RX, CRC generator and checker, central FSM) was analyzed in Quartus Prime 20.1 as the reference.

V. RESULTS

A. Existing UART Reference

Table I shows the timing result of the existing UART with CRC-based error detection in Quartus Prime 20.1. The analysis reports a setup timing violation with a worst-case slack of -2.569 ns on a data path delay of 3.348 ns, so the existing implementation does not meet its timing constraint. The CRC logic and its register path therefore need optimization, and the design should be re-analyzed after optimization to confirm that the slack is non-negative.

TABLE I. Timing Analysis of the Existing UART with CRC (Quartus Prime 20.1)

Parameter	Value
Worst-case setup slack	-2.569 ns
Data path delay	3.348 ns
Timing status	Violated

The proposed design adds LFSR scrambling to this CRC-protected path. Its Vivado synthesis, power and timing results are given in Section V-B (Figs. 3 to 7 and Tables III and IV). Table II summarizes the capability gaps in existing work that the proposed design addresses.

TABLE II. Capability Comparison of Existing Work and the Proposed Design

Work	CRC	Scramble	Low area	Security
UART + parity [7]	No	No	Yes	No
UART + FIFO [8]	No	No	No data	No
UART + CRC-16 [6]	Yes	No	Yes	No
ASCON UART [5]	No	No	No	Strong
Proposed	Yes	Yes	Yes	Light

B. Vivado Implementation Results of the Proposed Design

The proposed secure UART was synthesized and simulated in Vivado 2025.1 for the Artix-7 xc7a35tcpg236-1 device. The synthesized schematic in Fig. 3 contains 28 cells, 25 I/O ports and 52 nets. It consists of the receiver (secure_rx) and the transmitter (secure_tx), with every port passing through an input or output buffer and one BUFG on the clock.

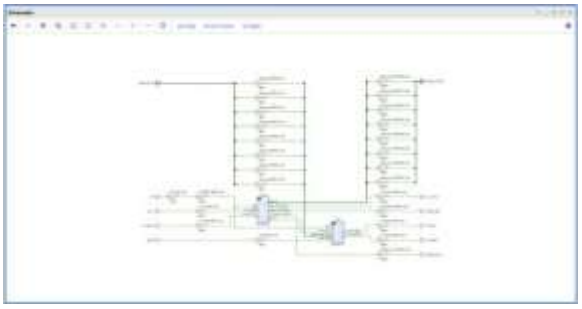


Fig 3: Synthesized Schematic of the Proposed Secure UART

The behavioural simulation of tb_secure_uart in Fig. 4 runs for 1 µs. The reset is released at about 100 ns and the send input pulses once, after which the 8-bit transmit data a5 is applied and the transmitter signals toggle. The flip input stays at 0, so no error is injected, and the data-valid, CRC-error and frame-error flags remain low in this window.



Fig 4: Simulation Waveform of the Secure UART Testbench

The Vivado power estimate in Fig. 5 and Table III gives a total on-chip power of 0.074 W, of which 0.070 W (95%) is device static power and 0.003 W (5%) is dynamic power. Clocks account for 63% of the dynamic power. The confidence level is Low because the switching activity is estimated without a simulation activity file.



Fig 5: Vivado Power Summary of the Proposed Design

TABLE III. Power Analysis of the Proposed Design (Vivado, xc7a35tcpg236-1)

Parameter	Value
Total on-chip power	0.074 W
Dynamic power	0.003 W (5%)
Device static power	0.070 W (95%)
Junction temperature	25.4 °C
Thermal margin	59.6 °C (11.9 W)
Effective θJA	5.0 °C/W
Confidence level	Low

The timing summary in Fig. 6 and Table IV shows that the design does not meet timing in this run. Setup has a worst negative slack of -0.112 ns on one of 424 endpoints, hold has a worst slack of -0.595 ns on 18 endpoints, and pulse width has no violation. All paths belong to the single sys_clk domain.



Fig 6: Design Timing Summary of the Proposed Design

TABLE IV. Design Timing Summary of the Proposed Design

Parameter	Setup	Hold	Pulse width
Worst slack	-0.112 ns	-0.595 ns	4.500 ns
Total slack	-0.112 ns	-10.405 ns	0.000 ns
Failing endpoints	1	18	0
Total endpoints	424	424	144
Status	Violated	Violated	Met

The intra-clock path report in Fig. 7 shows that the only failing setup path runs from u_tx/shreg_reg[3] to the tx_serial output, with a slack of -0.112 ns and a total delay of 5.674 ns (3.460 ns logic and 2.214 ns net) against a 10.0 ns requirement. The path from the transmitter state register to tx_busy meets timing with 0.620 ns of slack, and the receiver paths to crc_error and data_out[0] to data_out[6] each have 1.546 ns of slack.

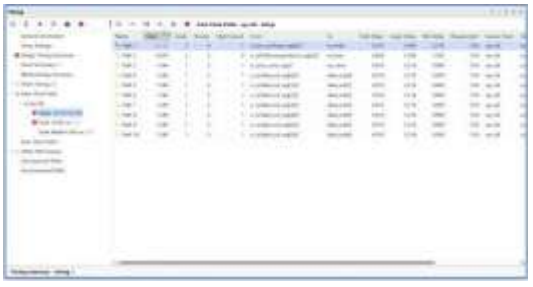


Fig 7: Intra-Clock Setup Paths sys clk of the Proposed Design

the simulation output of the fault-tolerant Universal Asynchronous Receiver Transmitter (UART) system integrated with CRC-16 error detection using Verilog HDL. The testbench transmits the input data 0xA5 through the UART transmitter, which is successfully received as 0xA5 by the receiver. The CRC error flag is displayed as 0, indicating that no data corruption has occurred during transmission. The transmitted and received data are identical, confirming the correct operation of the UART communication and CRC verification modules. The final simulation result is displayed as PASS, demonstrating successful data transmission, reception, and error detection functionality.



Fig 8: Simulation Results of the Proposed Design

VI. CONCLUSION

The proposed low-power secure UART provides reliable and lightly secured serial communication. LFSR-based scrambling randomizes the transmitted bit pattern, CRC error detection identifies corrupted frames, and the UART transmitter and receiver provide framing, synchronization and data recovery. The architecture uses only shift registers, XOR gates and finite-state logic, so it has low hardware complexity and suits VLSI implementation. Its power, area and delay remain to be measured by synthesis and simulation, and the timing violation found in the existing CRC UART must be resolved. LFSR scrambling is obfuscation only and is not a substitute for strong encryption.

VII. FUTURE SCOPE

The design can be extended with CRC-32 or configurable CRC polynomials for stronger detection, and with longer LFSRs, dynamic seeds or multiple LFSRs (or NLFSR elements) to resist keystream prediction. Clock gating and power gating can reduce dynamic and leakage power, and higher baud rates, configurable data formats and further area and delay optimization in advanced VLSI technologies can be explored.

REFERENCES

[1] Buddhanoy et al., “Extracting Data-Scrambling Keys from Commercial Off-the-Shelf (COTS) NAND Flash Memory,” 2026. [Venue details to be completed].

[2] K. R. Raghavendra, H. S. Manjunath, K. B. Hegde and M. B. Veena, “Enhanced LFSR-Based Frequency Divider with Multi-Stage Feedback Selection and State Extension,” Proc. 4th IEEE Int. Conf. Knowledge Engineering and Communication Systems (ICKECS), DOI: 10.1109/ICKECS70176.2026.11527966.

[3] P. S. Kartik, P. Muthukumar and L. Padmasuresh, “Comparative Performance Analysis of LFSR Techniques Using FPGA for Random Number Generation,” Proc. 9th Int. Conf. Circuit, Power and Computing Technologies (ICCPCT), 2026, pp. 2067-2071, DOI: 10.1109/ICCPCT70290.2026.11654872.

[4] J. Deng, Y. Zhang, H. Cui, H. Yang, D. Zhou, Y. Chen and X. Chen, “Lightweight PUF Authentication Protocol with NLFSR-Based Obfuscation,” Proc. Int. Conf. Electronics, Information, and Communication (ICEIC), DOI: 10.1109/ICEIC69189.2026.11386436.

[5] S. Y. Hatmojo, M. D. K. H. Yuntanozra and D. Ogi, “Secure Firmware Update over UART Using AEAD-ASCON: A Lightweight Cryptographic Framework for Constrained IoT Devices,” Proc. IEEE Int. Conf. Industry 4.0, AI, and Communications Technology (IAICT), Jul. 2026, pp. 629-636, DOI: 10.1109/IAICT71158.2026.11620842.

[6] P. B. Swetha, A. Gayathri, B. Renuka, D. Shirisha, C. S. R. Vennakandla and S. F. Sultana, “Fault-Tolerant UART with CRC-Based Error Detection,” Proc. 7th Int. Conf. Mobile Computing and Sustainable Informatics (ICMCSI), 2026, pp. 1684-1689, DOI: 10.1109/ICMCSI67283.2026.11412734.

[7] S. Karthigaa, M. Premalatha, S. Yuvikaa Shree, V. Jayanthimani, J. M. Keshika and M. S. Vinotheni, “Design and Simulation of UART with Configurable Parity,” Proc. 5th Int. Conf. Communication, Computing and Electronics Systems (ICCCES), 2026, , DOI: 10.1109/ICCCES62661.2026.11436434.

[8] S. Dhanasekar, A. K. Boopathy, A. V. P. Kumar, G. V. Amirth and D. Madhumitha, “Design of Low Power UART Architecture with FIFO Buffering for High-Speed Applications,” Proc. Int. Conf. Signal Processing and Electronics Design (ICSPED), 2026, pp. 182-188, DOI: 10.1109/ICSPED65817.2026.11448415.

[9] E. S. Rajavel, V. G. Rajkumar, M. S. Kashwin, V. J. Suthahar, A. Pravin and D. J. J. D. Daniel, “UART Design and Verification of Compatible Baud Rate Communication,” Proc. Third Int. Conf. Networking and Communications (ICNWC), 2026, DOI: 10.1109/ICNWC68145.2026.11518143.