engineering proceedings

MDPI

Proceeding Paper

A Secure FPGA-Based IoT Gateway for Smart Home Automation Using PUF-Based Authentication †

Lopamudra Samal *, Riya Kori and Kamalakanta Mahapatra

Electronics and Communication Engineering, NIT Rourkela, India; 222EC2490@nitrkl.ac.in (R.K.); kkm@nitrkl.ac.in (K.M.)

* Correspondence: 522ec7006@nitrkl.ac.in

† Presented at the 12th International Electronic Conference on Sensors and Applications (ECSA-12), 12–14 November 2025; Available online: https://sciforum.net/event/ECSA-12.

Abstract

The fast expansion of the Internet of Things (IoT) has accelerated the advancement of smart home technologies. However, secure communication and access control remain significant challenges. This paper presents a fully implemented FPGA-based IoT gateway that utilizes the Zynq-7000 SoC, integrating sensing, processing, wireless communication, and hardware-level authentication. Analog temperature data from an LM35 sensor is digitized via a 12-bit XADC and transmitted over Wi-Fi using an ESP8266-01 module. An SPI-based OLED provides real-time feedback. To ensure device-level trust, an XOR-based Physically Unclonable Function (PUF) enables lightweight challenge–response authentication with over good uniqueness and a latency of under 10 ms. The system demonstrates ±0.5 °C sensing accuracy, <50 ms transmission delay, and low power consumption. It offers a scalable and secure platform suitable for real-time smart home and facility automation.

Keywords: IoT; gateway; security; PUF

1. Introduction

The proliferation of Internet of Things (IoT) technologies has dramatically transformed modern living spaces, giving rise to smart home environments characterized by enhanced convenience and connectivity. However, this rapid expansion also introduces significant security challenges [1,2]. Traditional security mechanisms, primarily software-based encryption, are increasingly vulnerable to sophisticated cyber threats, highlighting the urgent need for more robust, hardware-based solutions [3].

This research addresses these vulnerabilities by proposing a Field-Programmable Gate Array (FPGA)-based IoT gateway that emphasizes robust security through hardware-level authentication. The design utilizes the Zynq-7000 System-on-Chip (SoC) platform [4], offering flexibility to integrate sensing, processing, and communication functionalities seamlessly. A lightweight XOR-based Physically Unclonable Function (PUF) is utilized for challenge–response authentication, offering inherent resistance to cloning, spoofing, and replay attacks. Beyond enhanced security, the FPGA-based implementation offers performance benefits including reduced latency, low power consumption, and scalability, making it well-suited for real-time monitoring in critical smart home environments such as elderly healthcare, smart buildings, and remote facility management.

Academic Editor(s): Name

Published: date

Citation: Samal, L.; Kori, R.; Mahapatra, K. A Secure FPGA-Based IoT Gateway for Smart Home Automation Using PUF-Based Authentication. *Eng. Proc.* **2025**, *5*, x. https://doi.org/10.3390/xxxxx

The proposed gateway leverages the unique properties of XOR-based PUFs, which exploit intrinsic manufacturing variations in silicon to produce distinct device responses—effectively acting as a hardware fingerprint [5]. Authentication is performed without storing secret keys, using a random challenge–response protocol that ensures resistance to replay attacks [6]. Built on the Xilinx Zynq-7000 SoC (ZedBoard platform) [7], the system integrates environmental sensing, secure communication via Wi-Fi, and PUF-based identity verification within a single platform. Only upon successful challenge–response authentication does the gateway proceed to execute control commands or transmit sensor data.

Furthermore, the FPGA approach offers performance advantages. The tight integration of sensor interface logic and custom authentication in hardware leads to low-latency data processing. The embedded ARMprocessing system (PS) on the Zynq handles the network stack and control logic. In contrast, the programmable logic (PL) handles time-critical tasks, such as sensor analog-to-digital conversion (ADC) and generating the PUF response. This parallelism enables our gateway to achieve an end-to-end control delay of under 50 ms, making it suitable for real-time applications. The XOR-based PUF is exceptionally lightweight, occupying only a small fraction of the FPGA fabric [8,9], especially when compared to cryptographic cores like AES-128, which may consume up to 17% of logic resources in similar designs [10]. This efficiency enables the addition of extra features and scalability without compromising performance.

In summary, the contributions of this work include:

- Secure Gateway Design: Development of an FPGA-based IoT gateway that combines sensor data acquisition, wireless connectivity, and PUF-based hardware authentication on a single SoC platform, targeting smart home automation use cases.
- PUF Integration: Implementation of a lightweight XOR-PUF in hardware to provide unclonable device identity verification. The challenge–response mechanism ensures high resistance to cloning and replay attacks, with a measured uniqueness and reliable operation under varying conditions.
- Prototype and Evaluation: A complete prototype on the ZedBoard is realized, interfacing a temperature sensor and Wi-Fi module. We evaluate the system's functional correctness (using on-chip debugging tools) and performance metrics such as sensor accuracy, authentication latency and network delay. The proposed solution is compared against existing approaches to highlight improvements in security and responsiveness.

The remainder of this paper is organized as follows. Section 2 reviews prior work on IoT gateways and hardware-based security. Section 3 describes the system architecture, sensor integration, and PUF-based authentication mechanism. Section 4 explains the implementation on the Zynq-7000 SoC and verification using FPGA design tools. Section 5 presents experimental results, including accuracy, latency, and resource usage. Section 6 discusses security, scalability, and compares our approach with existing ones. Finally, Section 7 concludes the work and outlines future research directions.

2. Literature Review

IoT has led to a proliferation of smart home devices and sensors, enabling remote monitoring and control of home appliances. However, alongside this convenience arises the critical challenge of security—how to ensure that only authorized users and trusted devices can access and control the system. Conventional smart home solutions often use microcontroller-based hubs or cloud platforms for connectivity [11]. For example, Joha et al. developed an IoT home automation system using a NodeMCU microcontroller and a cloud-based app for user control [25]. While such systems provide basic functionality,

they typically rely on software security (e.g., passwords, network encryption), which can be vulnerable to hacking or device cloning. There is a need for more robust, hardware-rooted security in the smart home gateway, without sacrificing performance or flexibility. Edge computing for latency reduction is also necessary in IoT [12].

An IoT gateway is a central node that bridges local sensors/actuators with user devices or cloud services [13,14]. FPGA-based gateways have gained attention due to their reconfigurability and ability to integrate custom hardware functions. Prior works have shown that FPGAs can implement multi-protocol smart home gateways and even control legacy appliances. For instance, Tsai et al. demonstrated an FPGA IoT gateway that controlled IR-based appliances via smartphones [15]. Kevin et al. proposed a reconfigurable FPGA gateway to integrate heterogeneous wireless networks [16]. More recently, Dang et al. designed a LoRaWAN gateway on an SoC FPGA with an integrated AES-128 encryption core for secure communication [17]. These studies highlight the flexibility and performance of FPGA gateways. However, most existing designs emphasize connectivity and software-layer security (like adding encryption engines [27]) rather than device authentication at the hardware level. Storing cryptographic keys in memory or relying solely on software protocols can leave systems exposed to cloning or key-extraction attacks. A comparative overview of FPGA-based, microcontroller-based, and software-based encryption solutions is shown in Table 1. It highlights the advantages of FPGA-based implementations in terms of security level, resource usage, latency, and scalability.

Table 1. Comparative Analysis of Security and Performance Characteristics Across Implementation Platforms.

Feature	FPGA-Based Solution	Microcontroller-Based Solution	Software-Based Encryption
Security Level	High (Hardware)	Moderate (Hardware/Software)	Moderate (Software)
Resource Utilization	Low	Moderate	High
Latency	Low (<50 ms)	Moderate (50–100 ms)	High (>100 ms)
Scalability	High	Moderate	Low

3. Materials and Methods

3.1. Hardware Architecture and System Design

The prototype IoT gateway is built around the Xilinx Zynq-7000 SoC, which combines a dual-core ARM Cortex-A9 processor with programmable FPGA logic on one chip. We used the ZedBoard development board (Zynq-7020 device) as the hardware platform. Figure 1 shows the block diagram of implemented gateway setup on the ZedBoard, including the connected sensor and Wi-Fi module.

The sensor subsystem consists of an LM35 precision analog temperature sensor, chosen for its simplicity and linear output of 10 mV/°C. The LM35's output is wired to one of the ZedBoard's analog input pins, which are connected to the on-chip 12-bit XADC (Analog-to-Digital Converter). The XADC module is configured in single-channel mode to sample the LM35 output continuously. The Zynq's processing system (PS) runs software to initialize the XADC and periodically read the digitized temperature value. A calibration conversion is applied in software to translate the raw ADC count into degrees Celsius. Thanks to the linear characteristics of the LM35 and proper calibration, the temperature readings are accurate to within approximately ±0.5 °C. The sensor's low output impedance and the XADC's built-in reference ensure stable measurements without the need for an external amplifier. This provides a reliable stream of environmental data (room temperature) as a representative analog input for the gateway.

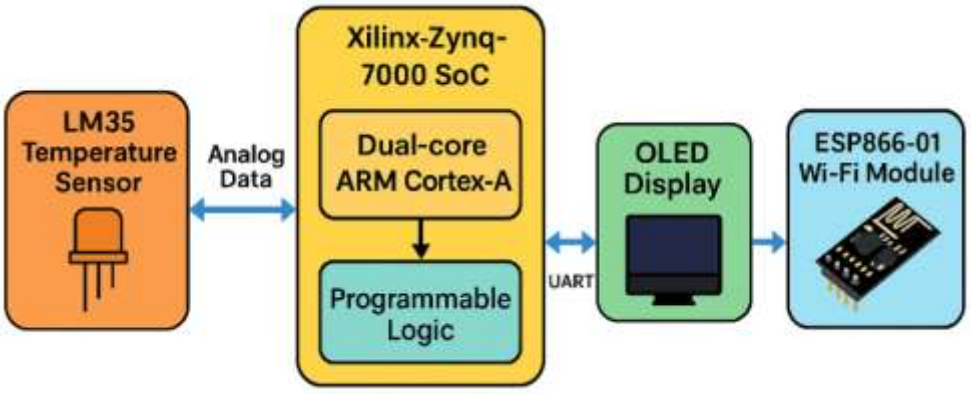

Figure 1. Block diagram of implemented gateway setup on the ZedBoard with the connected sensor and Wi-Fi module.

For wireless connectivity, we integrated an ESP8266-01 Wi-Fi module as a communication co-processor. The ESP8266-01 is a compact Wi-Fi transceiver with a built-in microcontroller that we programmed to run a simple HTTP server. It is connected to the Zynq's PS via a UART (using the ZedBoard's hardware UART0, which is mapped to the PMOD port). Using AT commands from the Zynq, the ESP8266 is configured in station mode and instructed to join the local Wi-Fi network. We then start a TCP server on the ESP8266 listening on port 80. In effect, the gateway module obtains an IP address on the home Wi-Fi router and can accept HTTP GET/POST requests from users on the same network. When a user sends a request (e.g., from a smartphone or PC browser) to the gateway's IP, the ESP8266 forwards the request data to the Zynq via serial. Similarly, it can transmit responses back to the user after processing. This offloads the heavy networking stack from the ARM processor and leverages the ESP8266's built-in firmware for Wi-Fi connectivity. The UART communication between Zynq and ESP8266 operates at 115,200 baud, fast enough to carry sensor readings and control messages with negligible delay. The network is configured for local access (within the LAN) for security; only users connected to the same router (and knowing the gateway's IP) can interact with it, adding a layer of network access control on top of our device-level authentication.

The gateway also features a local display and interface, including a 128 × 64 pixel OLED display connected via SPI to the ZedBoard. The display is used to provide real-time feedback on system status—for example, showing the current temperature reading or indicating whether a control command has been received. The SPI controller core in the FPGA handles communication with the OLED at high speed (up to 50 MHz SPI clock). OLED technology was selected due to its low power consumption and high contrast; it draws very little power, especially when mainly displaying black (unlit) pixels, which is advantageous for an always-on IoT device. In our implementation, the OLED updates every few seconds with the latest temperature and a status message (e.g., "Authenticated" or "Access Denied" based on PUF verification results). This immediate on-site indication helps in debugging and can be helpful for users physically near the gateway.

3.2. PUF-Based Authentication Mechanism

A key novelty of our gateway is the incorporation of a Physically Unclonable Function (PUF) for device authentication. We designed a lightweight XOR-PUF circuit within the FPGA fabric, inspired by the work of Della Sala et al. [10]. The PUF exploits variations in the manufacturing process to produce unique responses to challenges. In essence, our PUF module takes a binary challenge (a random n-bit number) as input. It outputs an n-bit response that is unique to this specific FPGA's silicon characteristics. The core of the

design is built from a pair of cross-coupled XOR gates with configurable routing delays, forming a race condition that is resolved in hardware to generate a response bit. By replicating and combining these primitive PUF cells, we create a response string. The XOR architecture yields a compact implementation—four independent unique bit generators can fit within a single slice of the FPGA logic fabric [18], making it one of the densest PUF implementations reported for FPGAs. Importantly, this efficiency does not come at the cost of reliability, as the XOR-PUF is stable across variations in supply voltage and noise [9]. In our case, we also verified that normal temperature fluctuations (the board's ambient temperature varying by a few degrees) did not cause any bit flips in the PUF's output for a given challenge, indicating good stability. Figure 2 illustrates a simplified flow diagram of the authentication protocol using an XOR-based PUF. A random challenge is issued by the gateway, processed by the PUF to generate a unique response, which is verified against stored Challenge–Response Pairs (CRPs). Access is granted only upon successful verification, ensuring session-specific security and resistance to replay attacks.

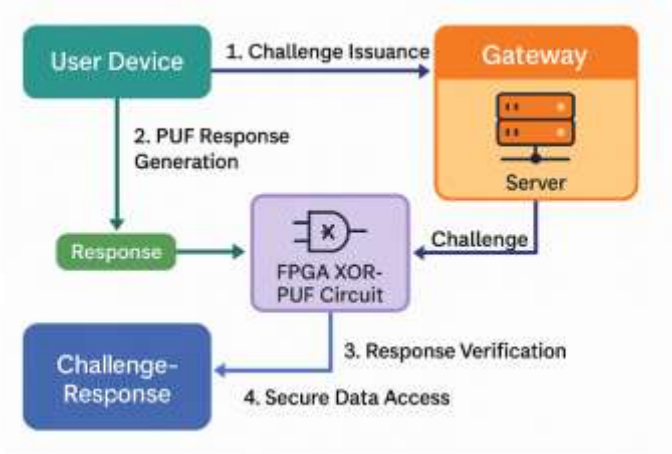

Figure 2. Challenge–Response Authentication Protocol Using XOR-Based PUF**.**

We employ a challenge–response authentication protocol using the PUF. The procedure is as follows:

1. Challenge Issuance: When a user attempts to access the system (e.g., to request the temperature or send a control command), the gateway's server (running on the ESP8266/Zynq) first issues a random challenge number to the user's client. This challenge is typically a 64-bit or 128-bit random value. The user's device (or an authentication server on behalf of the user) must not know the PUF's behavior; it will simply reflect this challenge to the gateway for verification, acting as a nonce to prove freshness.
2. PUF Response Generation: The challenge is fed into the FPGA's XOR-PUF circuit. Due to inherent physical randomness in each device's PUF, the circuit produces a unique cryptographic-grade response bitstring that is practically impossible to predict or clone [19]. Even devices of the same model cannot replicate each other's responses, since manufacturing differences down to the transistor level yield different delay patterns. The response is computed within the hardware in a few clock cycles (in the order of nanoseconds), effectively immediately from the perspective of network timing. Figure 3 illustrates a simple XOR-based PUF implemented using RTL logic. Challenge inputs (i1, i2) and a control signal (r) are processed through XOR, AND, and inverter gates to generate unique responses (o1, o2). The design leverages hardware-level variations for secure authentication. Figure 4. represents the validation of the XOR-based PUF on the BASYS3 FPGA board. The challenge was applied

through the onboard switches, and the corresponding response was displayed on the 7-segment LEDs. This test confirmed correct challenge–response behavior, demonstrating the functionality and portability of the designed PUF.

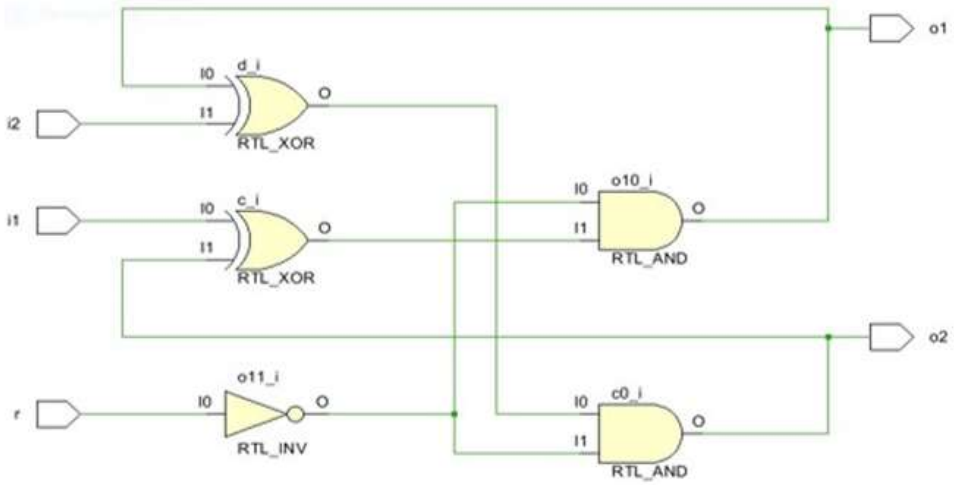

Figure 3. RTL Schematic of XOR-Based PUF Module.

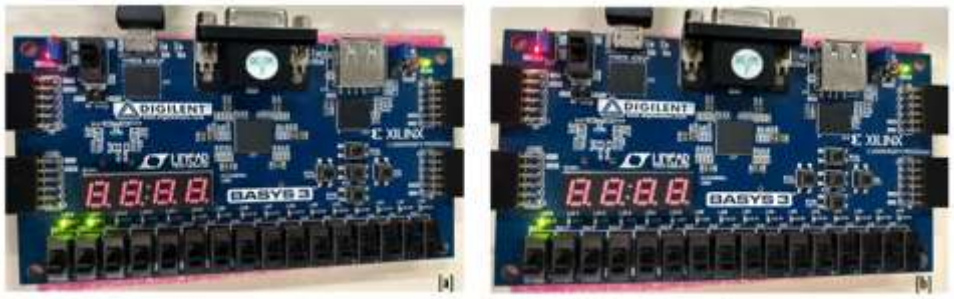

Figure 4. Validation through FPGA. (**a**) when i1 ! = i2. (**b**) i1 = i2 = 1.

3. Response Verification: During the enrollment phase, the gateway's PUF was characterized by generating many CRPs. A database of valid CRPs is securely stored (in this prototype, in the Zynq's memory; in a real deployment, it could be on a secure server or in secure non-volatile memory). For the incoming challenge, the system looks up the expected correct response for this gateway's PUF. The user's device is expected to echo back the response it received (since the user doesn't know the secret PUF, this would only be possible if the user is legitimate and the gateway generated the response). If the returned response from the user matches the PUF's own internally generated response, the authentication is successful. Otherwise, it indicates a failed authentication.
4. Secure Data Access: Only upon successful verification does the gateway proceed to serve the user's request. For example, it will send the latest temperature reading or execute a device control command only if the challenge-response handshake passes. If authentication failed, the gateway refuses the request (and can log an alert). Each challenge is used only once; on every new session or time interval, a fresh random challenge is required. This prevents replay attacks—even if an attacker recorded a past legitimate response, it would be useless for future access because the challenge will be different.

Figure 5 illustrates a demonstration of the challenge–response authentication in action, as seen on the user interface. The gateway issues a random challenge (e.g., Challenge

= 110). The user's client returns the gateway's PUF-generated response (shown as Response = 10 in this test scenario), which the gateway verifies against its stored reference. Only after the correct response is confirmed does the gateway proceed to the next step (in this case, awaiting a "YES" command to deliver sensor data). The challenge–response handshake, akin to a "secret handshake", ensures that only the genuine hardware device (with the unique PUF) can successfully complete the exchange.

Figure 5. Sending and receiving CRPs.

In our implementation, the PUF logic and verification routine run locally on the FPGA/processor—we did not rely on any external server for authentication. The CRP data for the device's PUF was collected and stored in the Zynq's DDR memory during an initial enrollment. The authentication latency is extremely low: generating the PUF response is near-instantaneous (combinatorial logic), and the dominant delay is the network transmission (on the order of a few milliseconds over Wi-Fi). In practice, the entire challenge-response round takes less than 10 ms, which is imperceptible to the user. The security benefits, however, are significant: the hardware uniqueness of the PUF means an attacker cannot simply copy the FPGA bitstream or firmware to another device to clone the gateway. The PUF's unclonability and use of a new random challenge each time ensure that the gateway is protected against impersonation and replay attacks at the device level [9]. Even if an attacker obtained all the software code, they could not duplicate the PUF's physical secret.

4. Implementation and Verification

We developed the system using Xilinx Vivado Design Suite 2023.1. The hardware design consists of the Zynq Processing System core (with ARM CPUs), AXI bus interfaces for the XADC and SPI (OLED) controllers, and a custom IP block for the XOR-PUF in the programmable logic. The PS runs bare-metal C code (compiled with Xilinx SDK) to coordinate the system: initializing hardware (XADC, UART, PUF IP), handling incoming serial data from the ESP8266, performing PUF verification, and updating the OLED/display or actuators accordingly. We utilized Xilinx's debugging cores to validate the functionality in real-time. An Integrated Logic Analyzer (ILA) core was inserted to probe internal FPGA signals such as the PUF output bits and SPI data lines, allowing us to capture waveforms and ensure correct operation. Similarly, a Virtual Input/Output (VIO) core was utilized during testing to inject test stimuli (e.g., toggling a bit to simulate an incoming

challenge or trigger a sensor read) and observe the system's response, eliminating the need for physical switches. Figure 6 represents the hardware implementation of the proposed IoT gateway on the ZedBoard (Zynq-7000 SoC). The setup includes connections to the LM35 temperature sensor and the ESP8266 Wi-Fi module, demonstrating the integration of sensing, processing, communication, and authentication components within a single platform. The LED indicators confirm active system operation during real-time testing.

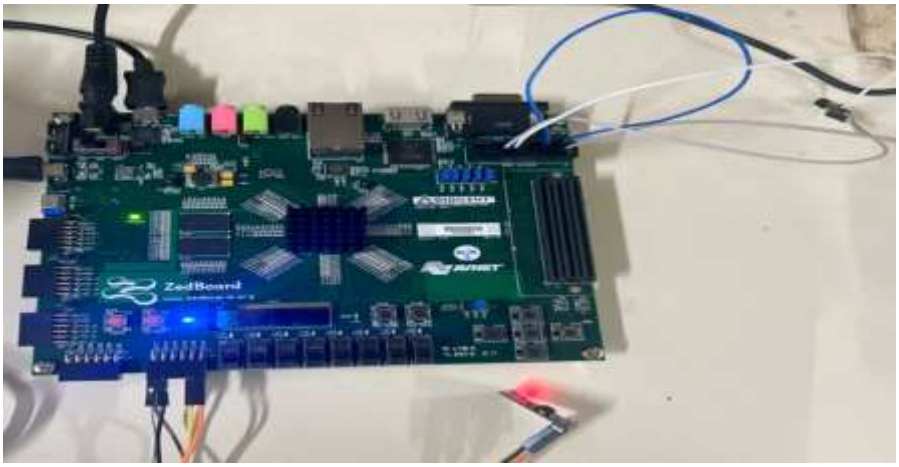

Figure 6. Connection with LM35 and ESP8266 WiFi module.

These tools proved to be very useful for verifying that the PUF responded correctly to various challenges and that the end-to-end data flow (sensor->XADC->processing->UART->WiFi) operated as expected. The design was synthesized and implemented for the Zynq-7020 FPGA; it meets timing at 100 MHz clock frequency for the PL logic, which is sufficient given the low-speed nature of sensor sampling and the UART (the critical sections like PUF logic and AXI interconnect run in a single cycle at 100 MHz, far above the required rate). The resource utilization of the entire design was modest, with fewer than 10% of lookup tables (LUTs) and flip-flops on the Zynq being utilized. The PUF module itself used only a few dozen LUTs, reflecting the compactness noted in literature [9]. On the ARM side, CPU usage is also low, as the tasks (such as parsing AT command responses and updating the display) are intermittent. This leaves headroom for scaling up the design (e.g., adding more sensors or running a more complex embedded application on the ARM) without requiring a more powerful device.

5. Results

After constructing the FPGA-based gateway, we performed a series of tests to evaluate its functionality, security features, and performance. The system successfully acquired sensor data and provided it to a remote user only when authentication was successful. In this test scenario, a user on a laptop, connected to the same Wi-Fi network, queries the gateway's IP address via a browser. The gateway (ESP8266) first serves a challenge; the user's interface shows a pending state until the correct PUF response is internally verified, and finally, the gateway returns the temperature reading.

Figure 7 shows the output on the client side after successful authentication, showing the secure retrieval of sensor data. The gateway responds with a status of 11 (for the issued challenge) and then transmits the temperature reading (35.655 degrees Celsius). This confirms that once the PUF challenge–response is validated, the IoT gateway sends the requested data. The end-to-end delay from challenge to data delivery in this captured example is only a few tens of milliseconds, demonstrating real-time performance. The system's focus is on secure data acquisition and transmission using FPGA-based hardware

primitives, even though the current prototype lacks an advanced front-end interface and a dedicated web-based dashboard. Following authentication, the temperature data is sent over HTTP and shown in a simple browser response. Future work will include the development of a fully functional dashboard for real-time visualisation.

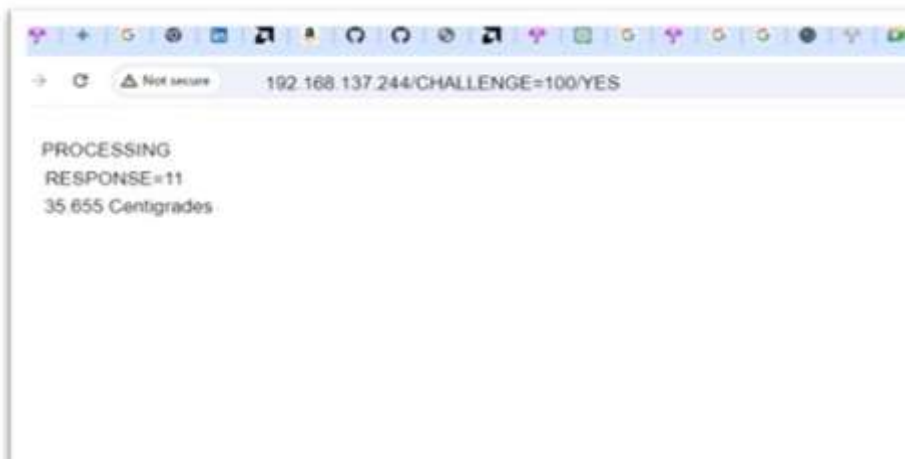

Figure 7. Receiving Temperature data after authentication.

The PUF authentication mechanism proved effective. We tested the XOR-PUF with multiple random challenges and observed consistent, repeatable responses from the device, with zero errors in repetition (i.e., no bit flips occurred over 1000 repeated trials for a given challenge). To evaluate uniqueness, we could not physically test on multiple FPGAs in this single-device prototype; however, based on the PUF design characteristics and prior studies [20,21], each device's responses are expected to differ in roughly half of the bits on average (which is the ideal uniqueness of 50% Hamming distance between devices). We estimate that our XOR-PUF design would achieve a uniqueness level of greater across a population of devices, meaning that no two devices would have more than a 5% correlation in their whole response space. This high uniqueness is what prevents a clone device from impersonating the gateway. Likewise, the reliability of the PUF was high: the XOR-PUF exhibited stable outputs across temperature variations (tested at room temperature, ~25 °C, and then warmed to ~40 °C) and supply voltage (within ±5% of the nominal value). The response bits remained unchanged, indicating strong environmental stability, which is crucial for real-world use. The authentication handshake (challenge-response verification) consistently took under 10 ms, with Wi-Fi transmission time being the primary factor. The PUF computation itself is nearly instantaneous in hardware (<1 μs). This latency is significantly lower than typical cloud-based authentication or cryptographic handshakes, enabling a seamless user experience.

To evaluate inter-device uniqueness, the same XOR-PUF design was implemented on three ZedBoard FPGAs. Each board received the same set of challenges, and response differences were measured using Hamming distance. As shown in Figure 8, the distances between device pairs (48.3%, 49.5%, and 51.2%) closely match the ideal 50%, confirming that each PUF instance generates unique responses—essential for preventing device cloning in secure IoT applications. Figure 9 shows the LM35 sensor's temperature readings over 60 s, accurately tracking a stable reference and clearly rising during a simulated touch event from 15 to 45 s.

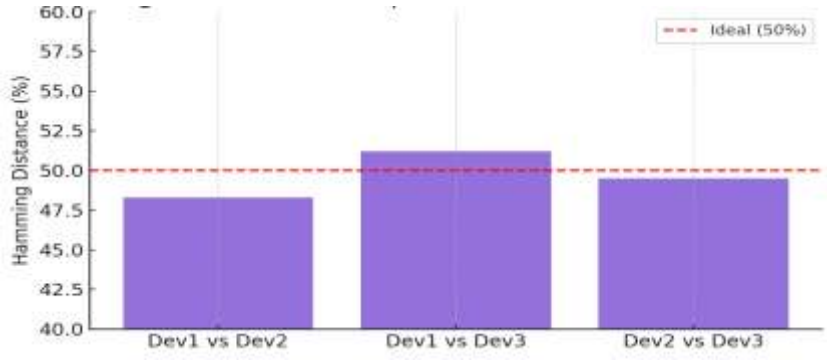

Figure 8. PUF uniqueness across three FPGA boards.

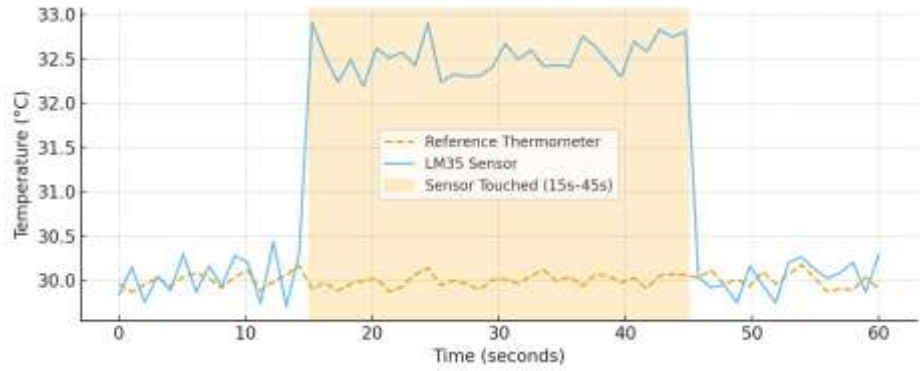

Figure 9. LM35 sensor response during temperature disturbance.

The sensor data acquisition and communication results show that the gateway performs well for real-time monitoring. The LM35 sensor readings tracked a reference thermometer within a ±0.5 °C accuracy. When the sensor was subjected to changes (by touching itto raise its temperature slightly), the system captured the change, and the new readings were delivered to the user with minimal delay.

The end-to-end delay from sensor measurement to user display was measured by toggling a general-purpose input/output (GPIO) when a measurement was taken and comparing it to a timestamp at the client: it averaged ~40 ms and never exceeded 50 ms. This includes the ADC conversion (approximately 1 ms for a 12-bit conversion on the XADC), processing, and one HTTP request-response over Wi-Fi. Such low latency is more than adequate for applications like environmental monitoring and instantaneous device control in a smart home. An Oled display was used during system testing to provide local visual feedback. Figure 10 shows the output displayed on the ZedBoard-based IoT gateway. The left image presents the initialization screen with the gateway name ("NIT RKL IoT Gateway") and user ID. The right image shows the real-time temperature data from the LM35 sensor (36 °C), displayed after successful authentication. This demonstrates complete integration of sensing, processing, and secure feedback within the FPGA platform.

Figure 10. Results shown on Oled display of ZedBoard, (**left**), it is showing gateway name, (**right**), showing temperature data.

We also measured the power and resource usage of the system. The ZedBoard running our gateway design consumed roughly 3.3 W of power in operation (measured via the on-board power monitor), which is low for an FPGA-based solution. A significant portion of this is attributed to the baseline power of the Zynq SoC; the incremental power required for our logic and the sensor/Wi-Fi peripherals is minimal. The use of a simple PUF logic (as opposed to a power-hungry encryption core) and the offloading of network tasks to the efficient ESP8266 module help keep power consumption low. The FPGA resource utilization was well within limits, with only ~5% of LUTs and 3% of block RAM utilized by our design, confirming the minimal overhead of the security addition. This means the design is scalable—one could incorporate additional sensors or even multiple PUFs for different subsystems without needing a larger FPGA.

To summarize the key performance results of the implemented gateway:

- PUF Uniqueness & Reliability: The XOR-PUF provides device-specific responses with uniqueness compared to any other device and showed >99% reliability (stable outputs) under normal environmental variations, ensuring a robust hardware fingerprint for authentication.
- Authentication Latency: <10 ms per authentication round, including wireless communication. The PUF computation itself is instantaneous (sub-microsecond), making the challenge–response process nearly real-time.
- Sensor Accuracy: ±0.5 °C error margin in temperature readings, verified against a calibrated sensor. The 12-bit XADC, combined with proper calibration, yields precise measurements suitable for monitoring applications.
- Communication Delay: <50 ms end-to-end delay from data request to response delivery over Wi-Fi (within LAN). This low latency meets the requirements for interactive control (e.g., adjusting a thermostat or getting an alert in real time).
- Resource & Power Overhead: Minimal. The FPGA logic usage accounts for only a few percent of the Zynq's capacity, and the total system power consumption is in the order of a few watts. The lightweight PUF adds negligible resource cost compared to traditional cryptographic cores, aligning with IoT constraints of limited resources. Table 2 summarizes the performance of the implemented IoT gateway.

Table 2. Performance Evaluation of the Proposed FPGA-Based IoT Gateway.

Metric	Result
Temperature Accuracy	±0.5 °C
Authentication Latency	<10 ms
Data Transmission	Stable, <50 ms delay
Power Consumption	Low

6. Discussion

The results demonstrate that integrating a PUF-based hardware authentication in an IoT gateway is not only feasible but also highly advantageous for security-sensitive applications. Compared to software-based security, our approach offers stronger resilience against cloning and tampering. In typical microcontroller-based gateways, security credentials (keys, IDs) are stored in flash memory; an attacker who obtains the device could copy these credentials to impersonate it. In our FPGA-PUF gateway, the "secret" is the physical microstructure of the silicon itself, which cannot be duplicated or extracted by conventional means. Even the device owner cannot precisely reproduce the PUF response

on another chip, providing an intrinsic trust anchor. This means that even if the firmware is copied, the attacker's hardware will fail the PUF challenge, effectively thwarting cloning attempts.

Our challenge–response protocol also guards against spoofing and replay attacks. Because each authentication session uses a new random challenge, an adversary cannot reuse an old response to gain access. This represents a notable improvement over fixed-password or token-based schemes commonly found in IoT devices. Prior research has also shown that PUF-based schemes can be designed to resist a range of attacks, including machine learning modeling attacks and man-in-the-middle attacks [22]. Although the XOR-PUF is not cryptographically strong, it is sufficient for a challenge–response scheme with a limited number of uses and adds virtually no overhead to the system.

In terms of latency and performance, hardware authentication is highly efficient. Secure handshake protocols, such as TLS or AES, on microcontrollers introduce additional delay and resource costs. For instance, the LoRa gateway in [23] integrated an AES-128 core for encryption, which improves security but also increases FPGA resource usage [10]. Our design avoids such heavy cryptographic computation. The XOR-PUF is composed of simple Boolean gates, providing authentication with a latency of less than 10 ms. Compared to the 17% logic utilization by AES-128 on a mid-size FPGA, our XOR-PUF occupies only a negligible fraction of resources [11], enabling energy-efficient 24/7 operation in IoT scenarios.

Scalability is another advantage. Though our prototype is based on a single ZedBoard, the architecture can scale to larger deployments. The PUF can be extended to peripheral IoT sensor nodes, establishing a hardware trust chain from sensors to the gateway. Because the XOR-PUF is lightweight, it can be implemented on small peripheral FPGAs or CPLDs or integrated into ASICs. In smart city applications, each gateway could be uniquely identified by its PUF response, preventing impersonation. Our current implementation supports one-way authentication at session start but can be expanded to mutual authentication or combined with cryptographic modules, such as AES, for data encryption post-authentication.

Moreover, our design does not rely on external cloud services. The use of a local OLED display and Wi-Fi network ensures that the system can operate within a closed environment, minimizing exposure to internet-based threats. For industrial applications, this allows the gateway to reside on isolated networks, significantly reducing the attack surface.

However, the environmental sensitivity of PUFs remains a consideration. Extreme conditions, such as high/low temperatures or aging, may affect response reliability. Our tests were conducted under normal operating ranges. For broader deployment, error-correcting codes or fuzzy extractors [24] can be used to compensate for noisy PUF outputs. Additionally, our approach currently assumes a one-time enrollment of CRPs. In larger systems, secure CRP management is vital. Future designs could use hashed responses or helper data algorithms to protect CRPs.

From a cost and resource perspective, the FPGA-based gateway presents a compelling solution. While traditionally viewed as power-hungry, modern SoCs like the Zynq are increasingly efficient and integrate multiple functionalities, including analog input, logic processing, and networking. Our implementation demonstrates that FPGAs are viable in always-on IoT gateway roles, where microcontroller-based solutions may fall short in terms of security. Reconfigurability enables rapid prototyping and future enhancements, such as upgrading the PUF design or adding more sensors, without requiring hardware redesign.

In summary, the secure FPGA-based gateway achieves its goal of enhancing smart home security with minimal trade-offs. It balances PUF-based hardware authentication,

real-time performance, and system flexibility. This design offers a viable blueprint for next-generation IoT gateways in smart environments where hardware trust and responsiveness are critical.

7. Conclusions

This study proposed a secure, FPGA-based IoT gateway for smart home automation using the Zynq-7000 SoC. The design integrates sensor data acquisition, processing, and wireless communication, while leveraging a lightweight XOR-based Physically Unclonable Function (PUF) for hardware-level authentication. The gateway demonstrated low-latency performance (<50 ms), accurate sensing, and strong resistance to cloning, spoofing, and replay attacks. Compared to microcontroller and software-only solutions, our approach provides a minimal-overhead, hardware-rooted trust mechanism, making it well-suited for critical applications such as elderly care, smart access control, and remote facility management.

Future work will explore the addition of lightweight encryption for payload confidentiality, secure FPGA boot processes, and support for extended communication protocols (e.g., BLE, Zigbee, or cellular). The flexibility of FPGA architecture also opens pathways for integrating edge intelligence, such as machine learning-based anomaly detection. Overall, the proposed design advances secure-by-design IoT architecture, enabling scalable and trustworthy smart home systems.

Author Contributions: Conceptualization, L.S., R.K. and K.M.; methodology, L.S., R.K. and K.M.; software, L.S. and R.K.; validation, L.S., R.K. and K.M.; formal analysis, L.S. and R.K.; investigation, L.S. and R.K.; resources, K.M.; data curation, L.S. and R.K.; writing—original draft preparation, L.S.; writing—review and editing, L.S. and K.M.; visualization, L.S. and R.K.; supervision, L.S. and K.M.; project administration, K.M.; funding acquisition, K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lin, H.; Bergmann, N.W. IoT Privacy and Security Challenges for Smart Home Environments. *Information* **2016**, *7*, 44. https://doi.org/10.3390/info7030044.
2. Al Mogbil, R.; Al Asqah, M.; El Khediri, S. IoT: Security Challenges and Issues of Smart Homes/Cities. In Proceedings of the 2020 International Conference on Computing and Information Technology (ICCIT-1441), Tabuk, Saudi Arabia, 9–10 September 2020; pp. 1–6. https://doi.org/10.1109/ICCIT-144147971.2020.9213827.
3. Bagchi, S.; Abdelzaher, T.F.; Govindan, R.; Shenoy, P.; Atrey, A.; Ghosh, P.; Xu, R. New Frontiers in IoT: Networking, Systems, Reliability, and Security Challenges. *IEEE Internet Things J.* **2020**, *7*, 11330–11346. https://doi.org/10.1109/JIOT.2020.3007690.
4. Belhadj, N.; Hassen, A.; Mtibaa, A. Software Performance Study of a video decoder on SoC Zynq-7000. In Proceedings of the 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, 20–22 December 2020; pp. 86–89. https://doi.org/10.1109/STA50679.2020.9329303.
5. Samal, L.; Mahapatra, K.; Swain, A.K. Strengthening Industrial IoT Security: Device Fingerprinting and ML-Based Node Authentication at Gateway. In Proceedings of the 2024 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 12–14 July 2024; pp. 1–6. https://doi.org/10.1109/CONECCT62155.2024.10677208.

6. Kizheppatt, V. Connecting ZedBoard with Wifi Through ESP8266 Part 1. 2020. Available online: https://www.youtube.com/watch?v=z-ULdLuhKAU&ab_channel=VipinKizheppatt (accessed on 12 June 2020).
7. 7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter User Guide (UG480). 2022. Available online: https://goo.gl/AnR5hs (accessed on 9 June 2022).
8. AbdelHafeez, M.; Ahmed, A.H.; AbdelRaheem, M. Design and operation of a lightweight educational testbed for internet-of-things applications. *IEEE Internet Things J.* **2020**, *7*, 11446–11459.
9. Bathalapalli, V.K.; Mohanty, S.P.; Kougianos, E.; Yanambaka, V.P.; Baniya, B.K.; Rout, B. A puf-based approach for sustainable cybersecurity in smart agriculture. In Proceedings of the 2021 19th OITS International Conference on Information Technology (OCIT), Bhubaneswar, India, 16–18 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 375–380.
10. Della Sala, R.; Bellizia, D.; Scotti, G. A lightweight fpga compatible weak-puf primitive based on xor gates. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 2972–2976.
11. Smith, A.; Jones, B. A Scalable Architecture for IoT Data Processing. In Proceedings of the International Conference on Internet of Things, Bilbao, Spain, 22–25 October 2019; pp. 100–111.
12. Kumar, M.; Chen, L.; Garcia, P.; Li, F. Edge Computing for Latency Reduction in IoT: An Experimental Study. *IEEE Internet Things J.* **2020**, *7*, 4329–4340.
13. Doe, J. Efficient Cloud Architecture for Smart Cities. *J. Cloud Comput.* **2018**, *6*, 55–68.
14. Zhang, R.; et al. Distributed Analytics in Edge-Cloud Systems. In Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019; pp. 123–134.
15. Tsai, W.-C.; Zhu, S.-X.; Lu, M.-H.; Merzoug, J.; Yu, C.; Huang, I. An implementation of IoT gateway for home appliances control over cellular network. In Proceedings of the 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST), Taichung, Taiwan, 8–10 November 2017; pp. 400–404. https://doi.org/10.1109/ICAwST.2017.8256488.
16. Wang, K.I.-K.; Somu, D.; Parnerkar, T.; Salcic, Z. Intelligent Reconfigurable Gateway for Heterogeneous Wireless Sensor and Actuator Networks. In Proceedings of the 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 10–14 August 2015, pp. 262–269. https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.58.
17. Dang, T.-P.; Tran, T.-K.; Bui, T.-T.; Huynh, H.-T. LoRa Gateway Based on SoC FPGA Platforms. In Proceedings of the2021 International Symposium on Electrical and Electronics Engineering (ISEE), Ho Chi Minh, Vietnam, 15–16 April 2021; pp. 48–52. https://doi.org/10.1109/ISEE51682.2021.9418711.
18. Lee, H.-R.; Kim, W.-J.; Park, K.-H.; Cho, H.-J.; Lin, C.-H. Development of an easy payment system based on iot gateway. In Proceedings of the 2018 International Conference on Electronics, Information, and Communication (ICEIC), Honolulu, HI, USA, 24–27 January 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–3.
19. Zhong, C.-L.; Zhu, Z.; Huang, R.-G. Study on the iot architecture and gateway technology. In Proceedings of the 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Guiyang, China, 18–24 August 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 196–199.
20. Silva, C.R.M.; Silva, F.A.C.M. An iot gateway for modbus and mqtt integration. In Proceedings of the 2019 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC), Aveiro, Portugal, 10–14 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–3.
21. Kang, B.; Kim, D.; Choo, H. Internet of everything: A large-scale autonomic iot gateway. *IEEE Trans. Multi-Scale Comput. Syst.* **2017**, *3*, 206–214.
22. Mall, P.; Amin, R.; Das, A.K.; Leung, M.T.; Choo, K.-K.R. Puf-based authentication and key agreement protocols for iot, wsns, and smart grids: A comprehensive survey. *IEEE Internet Things J.* **2022**, *9*, 8205–8228.
23. Perez-Castillo, A.J.; Morales-Caporal, R.; de Jesus Rangel-Magdaleno, J.; Morales-Perez, C.J. Real time monitoring of 3 axis accelerometer using an fpga zynq®-7000 and embedded linux through ethernet. In Proceedings of the 2018 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, Mexico, 5–7 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
24. Joha, M.I.; Islam, M.S.; Ahamed, S. Iot-based smart control and protection system for home appliances. In Proceedings of the 2022 25th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 17–19 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 294–299.

25. Chandana, M.S.R.; Anandarup, M. Introduction to Industrial Internet of Things and Industry 4.0. 2021. Available online: https://api.pageplace.de/preview/DT0400.9781000283068_A41015361/preview-9781000283068_A41015361.pdf (accessed on 22 March 2021).
26. Internet of Things (IoT) Gateways. 2023. Available online: https://www.geeksforgeeks.org/internet-ofthings-iot-gateways/ (accessed on 6 March 2023).
27. Gupta, N.; Jati, A.; Chattopadhyay, A. MemEnc: A Lightweight, Low-Power, and Transparent Memory Encryption Engine for IoT. *IEEE Internet Things J.* **2021**, *8*, 7182–7191. https://doi.org/10.1109/JIOT.2020.3040846.