IRIS: Enhancing the Security of IoT Devices Using Internal IR-Based Sensors

Amit Kama[a,*], Yarin Kalfon[a], Yossi Oren[a]

[a]*Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beersheba, 8410501, Israel*

Abstract

Authentication in Internet of Things (IoT) environments faces significant challenges due to the devices' limited security capabilities and operational constraints, such as reduced computational power and energy. The unsecured and diverse settings in which these devices operate further complicate the implementation of traditional authentication protocols. While some work has explored leveraging intrinsic variations in Static Random-Access Memory (SRAM) characteristics for authentication, relatively little attention has been given to authentication approaches based on other sensors. In this work, we survey sensors commonly found in IoT devices and assess their suitability for authentication purposes. We identify the infrared (IR) receiver as a promising candidate for authentication, and demonstrate a practical method for using the inherent physical variations in these sensors to authenticate IoT devices. Our results demonstrate that IR receivers can authenticate IoT devices with an average accuracy of 0.9855, with a standard deviation of 0.014, above a base rate of 0.05. Motivated by these findings, we developed IRIS, a novel IR-based Identification System, and made an open-source artifact repository available to support further research. We also demonstrate the robustness of our proposed method under various constraints, such as shorter trace lengths, reduced sampling frequencies, relying solely on the receiver's data, and authenticating with a TV remote control. Our findings suggest that low-cost sensors like IR receivers can significantly enhance IoT devices security without increasing their cost.

*Corresponding author

Email address: kamaa@post.bgu.ac.il (Amit Kama)

Keywords: IoT Authentication, Physical Security, Physically Unclonable Functions, Fingerprintable Sensors, Infrared.

1. Introduction

Authentication is a critical security process used to verify the identity of users or devices before granting access to systems, resources, or information. It serves as the first line of defense in protecting sensitive data and ensuring that only authorized entities can interact with a system. Authentication methods can range from simple password-based mechanisms to more sophisticated approaches like biometric verification, two-factor authentication, or digital certificates [1]. These techniques assess the credibility of a claimed identity based on something the user knows (like a password), something the user has (such as a security token), or something the user is (like a fingerprint) [2]. Effective authentication practices are essential for maintaining the integrity and confidentiality of a system, and are foundational to a comprehensive security strategy.

One of the areas where authentication is particularly challenging is the Internet of Things (IoT) ecosystem [3]. In the IoT ecosystem, authentication takes on additional complexity due to the sheer number and diversity of connected devices. These edge devices, ranging from consumer electronics such as smart home appliances and entertainment systems to industrial sensors and actuators, operate on the periphery of the network and often handle critical functions or sensitive data. Ensuring that each device is properly authenticated is crucial to safeguard against unauthorized access and potential security breaches. This is particularly challenging given that edge devices frequently operate in unsecured environments and may have limited computational power to handle complex authentication protocols. As such, IoT security strategies must incorporate lightweight, robust authentication mechanisms that can operate efficiently on resource-constrained devices while providing a level of security that matches the risk associated with the device's function and the data it processes [4, 5, 6].

Traditional authentication methods, such as passwords, personal identification numbers (PINs), and cryptographic keys, are commonly used in many digital systems to verify identity. These methods rely on storing sensitive credentials and using encryption algorithms to protect them. While effective in traditional computing environments, these approaches are problematic for IoT. Passwords and PINs, for example, are vulnerable to brute

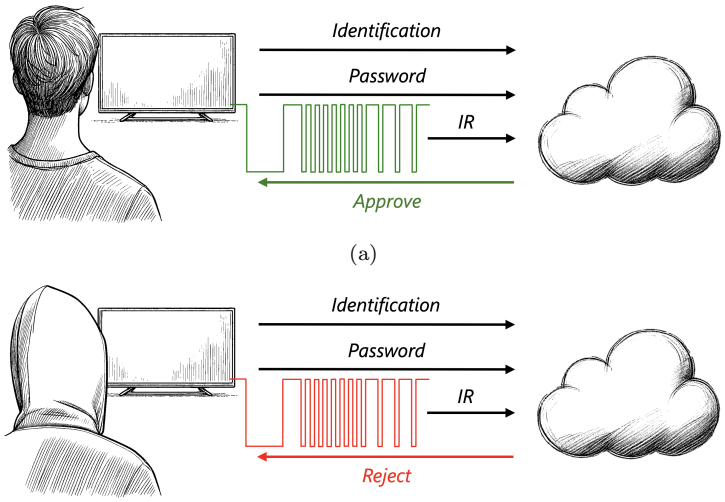


Figure 1: (a) A benign user attempts to authenticate against a cloud server. The server selects a challenge, reads the response, and verifies its correctness using a pre-trained model. (b) A malicious attacker tries to authenticate with the server, impersonating the benign user. Since the attacker's response to the challenge does not match the legitimate user's, the authentication fails.

force attacks and can be difficult to manage across a large number of devices. Cryptographic key-based systems, although secure, often require significant computational resources, which many IoT devices lack [7]. Furthermore, the need for secure storage of these keys introduces additional challenges for devices with limited memory [8]. Given the scale and resource constraints of IoT devices, relying on traditional methods makes it difficult to achieve secure, efficient, and scalable authentication across the entire network. This necessitates the development of alternative approaches tailored to the unique requirements of IoT systems.

One promising direction for IoT authentication is leveraging the inherent physical variations present in hardware components. These variations arise naturally during the manufacturing process, creating slight but consistent differences between seemingly identical devices. These differences are difficult to replicate and can serve as unique fingerprints for each device [9, 10]. Exploiting these physical characteristics allows for the development of authentication methods that are both secure and resource-efficient, making them ideal for IoT environments where traditional approaches may fall short [11].

Exemplifying this approach are Physically Unclonable Functions (PUFs), which use these random physical variations to generate a unique identifier for each device. PUFs, and in particular silicon PUFs, capitalize on the inherent unpredictability of semiconductor manufacturing processes to create secure and unclonable identifiers [12, 13]. Because PUFs require minimal additional hardware and leverage existing chip characteristics, they provide a cost-effective and scalable solution for robust security. This makes PUFs particularly well-suited for IoT devices, where adding sophisticated security hardware is often impractical [14]. Moreover, PUF-based authentication systems can effectively prevent a wide range of attacks, such as counterfeit device insertion and replay attacks, thereby enhancing overall network security [15, 16].

In recent years, an increasing variety of PUF constructions have been suggested for authenticating edge devices in the IoT domain. For example, SRAM-PUFs, as defined by Cortez *et al.*, take advantage of the way Static Random-Access Memory (SRAM) acts on startup. As each SRAM cell has its own bias toward a preferred startup value (one, zero, or random), a unique fingerprint can be created using multiple SRAM cells' startup values [17]. In [11], Chatterjee *et al.* presented RF-PUF, a deep neural network framework for real-time authentication of wireless nodes that exploits process variation effects on transmitter Radio Frequency (RF) properties, which are analyzed using in-situ machine learning at the receiver end. In [18], Yunmok *et al.* proposed GyrosFinger, a fingerprinting method that binds a drone's identity to its location by utilizing the unique offsets of micro-electromechanical systems (MEMS) gyroscopes, which are crucial for maintaining the drone's attitude and vary due to manufacturing discrepancies.

Many IoT devices contain components whose physical variations can be leveraged for security solutions [19]. However, to date, no comprehensive survey has been conducted to systematically evaluate which sensors and components in IoT devices are most suitable for authentication purposes. To address this gap and further advance authentication in IoT, we propose a more systematic approach to identifying and evaluating these components. This approach involves reviewing various hardware components commonly found in IoT devices, identifying those most suitable for authentication, and highlighting underexplored components that hold potential for developing sustainable security solutions.

Our comparative analysis of various hardware components, as discussed in section 2, led us to the understanding that the infrared (IR) receiver

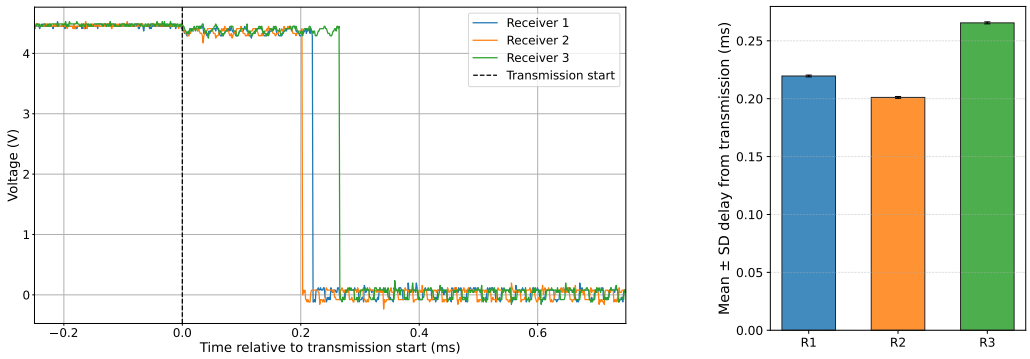
is a particularly strong candidate for fingerprinting due to its unique combination of ubiquity, highly-variable analog components, and ease of integration. IR receivers are widely used in IoT devices, particularly in smart home systems, industrial control interfaces, and remote-controlled consumer electronics, making them a practical target for authentication-based applications [20, 21, 22]. Despite their promising fingerprinting properties, their potential for robust authentication remains largely untapped. In this study, we aim to address this gap by showing how to integrate IR receivers into an innovative authentication protocol without incurring additional hardware costs for edge devices. The key factor enabling this approach is the inherent non-uniformity in the hardware of different IR receiver units, which affects their practical reception performance.

As a motivating example, we observed that multiple units of the same IR receiver model exhibited distinct delays from the transmission start to their first logical transition. To ensure that these differences are not simply due to varying transmission distance or placement, all receivers were positioned identically with respect to the transmitter. As shown in Fig. 2a, these device-specific timing offsets are clearly visible when traces are aligned to the transmission start. Fig. 2b quantifies the effect, showing that the delays remain consistent across 100 transmissions per device, confirming that IR receivers differ systematically in how they process signals even under controlled conditions.

Building on these insights, we propose a novel IR-based Identification System (IRIS). Our contribution focuses on demonstrating the feasibility of leveraging IR receiver variations for device authentication and designing a practical, deployable system. In this system, the authenticator issues a challenge to the user in the form of an IR transmission. By analyzing the IR receiver's unique response to this challenge, the prover can differentiate between devices and enable robust authentication.

We evaluate IRIS in a controlled lab setting using a dataset comprising 20 units of a common IR receiver. To further assess its real-world applicability, we test IRIS under various deployment conditions, including authentication using a TV remote control. Our results show that IRIS can identify devices with an average accuracy of 0.9855, with a standard deviation of 0.014, significantly surpassing the base rate of 0.05, all without requiring additional hardware on the authenticating IoT device.

To summarize, this paper makes the following contributions:



(a) (b)

Figure 2: (a) Example traces from three units of the same IR receiver model, aligned to the transmission start (dashed line). To rule out placement effects, all receivers were positioned identically relative to the same IR transmitter, yet distinct device-specific timing offsets are clearly observed. (b) Quantitative summary of the delay from transmission start to the first receiver falling edge, averaged over 100 transmissions per device. Results confirm consistent device-specific offsets (R1: 0.2196 ± 0.0008 ms, R2: 0.2011 ± 0.0008 ms, R3: 0.2655 ± 0.0009 ms).

1. We surveyed sensors commonly used in IoT and identified IR receivers as promising candidates for authentication (section 2 and table 1).
2. We developed a highly replicable lab setup to capture variations in IR receivers and provided an open-source artifact repository to facilitate further research (section 4.1).
3. Using data collected from this setup, we demonstrate that individual IR receivers can be authenticated by analyzing their signal outputs (section 5).
4. We address the challenges of transitioning the system from lab conditions to real-world scenarios by showcasing its robustness under various constraints, including shorter trace lengths, reduced sampling frequencies, reliance solely on the receiver's data, and authentication using a TV remote control (section 5).

We believe that our proposed method can be used in IoT deployments that contain IR receivers, significantly increasing their security without increasing their cost. Implementing IRIS in IoT devices could provide them with a substantial security benefit by offering a robust authentication method which is both cost-effective and scalable, since it utilizes the devices' intrinsic characteristics for security. By making IoT devices more difficult to dupli-

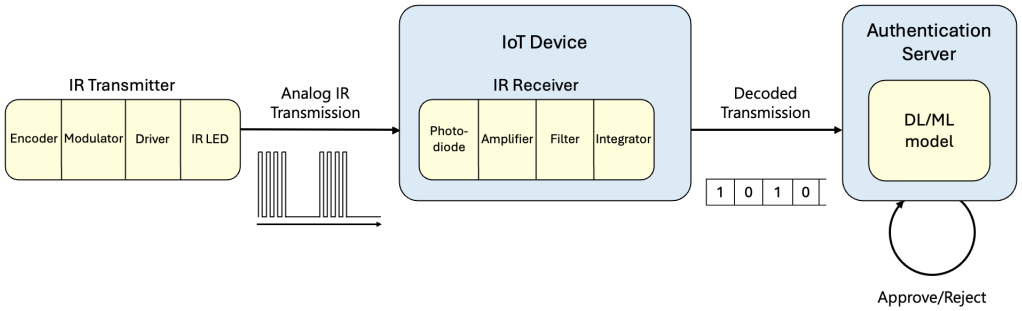


Figure 3: High-level block diagram of the IR-based identification system. An IR transmitter sends an analog signal to an IoT device, where the embedded IR receiver captures and processes it via its internal stages–photodiode, amplifier, bandpass filter, and integrator–whose device-specific variations form the basis of our method. The decoded transmission is then forwarded to an authentication server hosting a DL/ML model, which outputs an approve/reject decision.

cate or forge, IRIS can make the entire network more resilient to attacks, increasing the overall trust in IoT networks and allowing them to be used in more industry sectors.

The IRIS open-source artifact repository is available at the following link: https://github.com/AmitKama/IRIS.

2. The Fingerprinting Surface of IoT Devices

2.1. Introduction

In addition to their general-purpose computation and communication capabilities, IoT edge devices often feature a diverse array of hardware components which let them monitor and interact with their surrounding environment. As a motivating example, let us consider one such common device, the smart thermostat. This device naturally includes a microcontroller unit (MCU) for processing, on-board memory for storage, and communication modules like Wi-Fi and Bluetooth for connectivity. In addition to these core components, the smart thermostat must include sensors which allow it to monitor temperature and humidity. Additionally, it typically contains an IR receiver to receive commands from a remote control, and possibly other components, such as light sensors, a microphone, and a display. Our key observation is that each of these components exhibits subtle hardware-level differences across devices due to manufacturing variations, environmental

conditions, and other factors [12]. These differences provide a unique *"fingerprinting surface"* that can be exploited for secure and reliable device identification.

In this section, we will explore the fingerprinting potential of hardware components that are commonly found in IoT devices. Later, we will perform a comparative analysis of these components, and highlight components with strong fingerprinting properties, which can be utilized for authentication.

2.2. Methods

To systematically evaluate which IoT hardware components hold strong fingerprinting potential, we began by defining the key properties that make a component suitable for device identification. While this comparative analysis is arguably qualitative, it was instrumental in pointing us towards the most promising components for further investigation.

A component with strong fingerprinting properties must exhibit *uniqueness, consistency,* and *scalability* [23, 24]. Uniqueness refers to the ability of the component to generate distinct signatures that vary sufficiently across devices. Consistency ensures that these signatures remain stable and reproducible over time. Finally, scalability indicates how well the fingerprinting solution can be extended to a large number of devices without losing its effectiveness. These properties form the foundation for evaluating the potential of various IoT components.

We selected key IoT components for this comparative analysis based on their prevalence in IoT devices. Based on literature review, each component was evaluated against the defined fingerprinting properties. While we did not conduct experiments on every IoT component, we critically analyzed their potential based on the available literature. This process resulted in identifying strong candidates for further experimental study, setting the stage for deeper exploration in the next phase of the research.

2.3. Results

Our initial exploration identified key categories of IoT components, which were selected to provide comprehensive coverage across commonly-used device types. This selection focuses on components with high prevalence in IoT applications, and spans five categories: processing units, memory and storage, communication modules, environmental sensors, and audio and visual components. The results of our analysis can be seen in Table 1, which outlines each component's potential relevance to unique device identification.

As shown in the table, many studies that explore the use of variations for authentication, particularly in PUFs, have focused on processing units as well as memory and storage components, with comparatively less emphasis on communication modules and peripheral sensors. This trend highlights an opportunity to explore these additional components.

The IR receiver, in particular, stands out due to its high prevalence across IoT devices, especially in consumer electronics, and its suitability for large-scale, cost-effective applications. Its widespread use in TV streamers, monitors, air conditioning units, and security systems — key components of modern consumer electronics — combined with its low cost, makes it a practical and accessible choice for large-scale fingerprinting in IoT [20, 21, 22]. Moreover, its distinctive characteristics, such as the unique signal variations resulting from a combination of analog and digital processing, position it well for capturing stable, device-specific fingerprints. Despite its promising fingerprinting properties, there is very limited representation in existing literature about the IR receiver's potential for device authentication. We thus chose IR receivers as the focus of our continued study.

Table 1: Comparative analysis of fingerprinting properties in common components of IoT devices

Category	**Component**	**Uniqueness**	**Consistency**	**Scalability**	**Examples of Fingerprinting Methods in IoT**
Processing Units	SoC	High	High	High	SRAM-PUF [25], RO-PUF [26], Arbiter PUF [27]
	MCU	High	High	High	SRAM-PUF [25], RC PUF [28], RO-PUF [26]
Memory and Storage	RAM	High	High	High	SRAM-PUF [25], DRAM-PUF [29, 30]
	eMMC	Medium	Medium	Medium	–
	Flash	High	High	High	Flash-PUF [31, 32]
Comm. Modules	Wi-Fi	High	Medium	High	RF-PUF [11]
	Bluetooth	High	Medium	High	RF-PUF [11], Ali *et al.* [33]
	IR receiver	High	High	High	–
Env. Sensors	Temperature	Medium	Low	High	Labrado *et al.* [34]
	Humidity	Medium	Low	Medium	–
	Light	Medium	Medium	Medium	Light Sensor PUF [35]
	Accelerometer	High	Medium	High	AccelPrint [36]
	Gyroscope	High	Medium	Medium	GyrosFinger [18]
Audio and Visual	Microphone	High	Low	Low	S2M [37], MicPrint [38]
	Image sensor	High	Low	Low	Cao *et al.* [39]

3. Background

3.1. Related Work

PUFs were first introduced by Pappu *et al.* [40] in 2002 as physical one-way functions. They leverage physical variations to generate unique, fixed-length binary strings for device identification and authentication. In [41], Gassend *et al.* expanded the PUF domain by introducing silicon physical random functions, designed to identify and authenticate integrated circuits (ICs) using inherent manufacturing variations. By leveraging the statistical delay variations in wires and components across different ICs, they developed a parameterized self-oscillating circuit for IC characterization, an implementation now commonly referred to as a Ring Oscillator PUF (RO-PUF). In addition to delay-based intrinsic PUFs, memory-based PUFs have also emerged. One notable example is a technique that measures the startup values of memory cells for digital fingerprinting, which later became known as the SRAM-PUF and is now widely adopted [25, 42].

Alongside the well-known SRAM-PUF, many recent studies have explored new PUFs to enhance IoT device security by leveraging unique manufacturing variations. In [43], Kumar *et al.* introduced the Butterfly PUF, a PUF designed for Field-Programmable Gate Arrays (FPGAs), utilizing cross-coupled latches within the FPGA matrix that behave like SRAM cells during startup, entering an unstable state before settling into one of two stable states based on intrinsic variations. In [11], Chatterjee *et al.* introduced RF-PUF, a deep learning-based framework for real-time authentication of wireless nodes, leveraging inherent process variations in RF transmitters. This approach uses existing RF communication infrastructure without requiring additional hardware for PUF generation or feature extraction, placing the identification task entirely on the receiver.

As mentioned in [18], the concept of using sensors as PUFs was initially introduced by Rosenfeld *et al.* [35], in the context of securing remote sensors. They introduced a sensor PUF that generates unique responses using light levels and challenge bits as inputs. Its design features an array of photodiodes coated with material that creates chip-specific optical variations, enabling device-specific responses. Since then, several works have explored the use of sensors as PUFs by leveraging their inherent variations. For instance, the unique offsets and drift characteristics of MEMS-based sensors (gyroscopes and accelerometers) have been utilized for PUF-based fingerprinting. One notable example is the work by Son *et al.* [18], which proposed

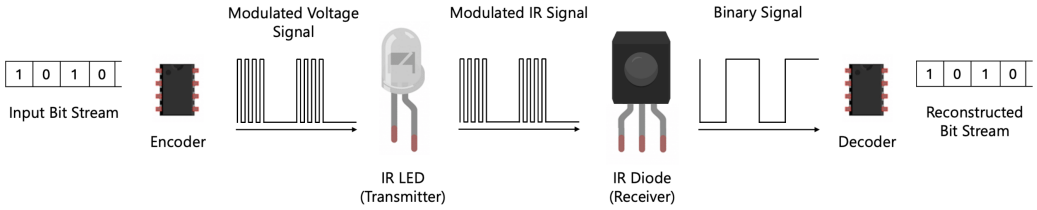


Figure 4: Overview of the IR communication process, illustrating how an input bit stream is encoded, transmitted via modulated IR signals, and decoded back into the original bit stream at the receiver [46] [47].

a fingerprinting method for drones in motion, that binds a drone's identity to its location using the unique offsets of MEMS gyroscopes caused by manufacturing differences. Experimental results show high F-scores (up to 98.78%) for distinguishing between 70 gyroscopes using the offsets from different axes, and the fingerprints remain consistent over time. That said, this work applies only to devices equipped with MEMS gyroscopes. Beyond that limitation, due to the unknown probability distribution of MEMS gyroscope offsets, the authors approximate entropy using assumptions about axis independence and uniform distribution. These assumptions may not hold for larger or different datasets, potentially leading to inaccurate entropy estimates and affecting the robustness and uniqueness of the proposed fingerprints.

While current sensor-based PUFs, such as those using MEMS sensors, have demonstrated promising results, their applicability remains limited.

3.2. Infrared Data Exchange

IR communication relies on infrared light waves to transmit data over short distances. An IR Light Emitting Diode (LED) outputs light in the infrared range, with typical wavelengths of between 850 nm and 950 nm. This light is further modulated with a pulse pattern, typically at a frequency of 38 kHz. The actual binary data is sent by switching this pulsed output on and off, with a typical bit rate ranging from approximately 500 bps to 1 Kbps, depending on the transmitted data. An IR receiver detects these pulses, demodulates the signal, and converts it back into binary data. Common IR protocols, such as the NEC infrared transmission protocol, use fixed pulse distance encoding and specify data formats for transmission and error checking [44, 45].

3.3. Structure of an IR Receiver

As illustrated in Fig. 5, a typical IR receiver processes the incoming infrared signal through a series of components designed to extract the transmitted data. The process begins with the photodiode (PD), which converts infrared light pulses into a weak electrical signal. This signal is initially amplified by a pre-amplifier, which provides high gain and low noise amplification to enhance the signal strength while minimizing noise. The signal is then passed through an amplifier for further boosting and a limiter to ensure the signal stays within a safe amplitude range. A bandpass filter, tuned to the carrier frequency of 38 kHz, isolates the relevant signal while filtering out ambient noise [48, 49].

After filtering, the signal is demodulated to extract the encoded information from the carrier wave. It is then processed by the integrator, which acts as a signal conditioner to enhance the reliability of subsequent binary classification. The comparator is configured as a Schmitt trigger to compare the input signal against a reference threshold, ensuring stable digital output and preventing rapid toggling caused by noise or small signal variations around the threshold. This ensures reliable detection of the original binary data, which is then ready for decoding and further processing by the receiving device [50].

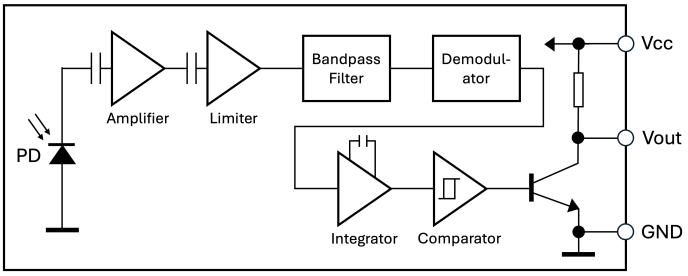


Figure 5: VS1838B IR receiver block diagram [48].

The physical manufacturing variations inherent in the components of an IR receiver, such as the photodiode, amplifier, bandpass filter, and comparator, can introduce subtle differences in the timing and behavior of signal processing. These variations arise from the imperfections and tolerances during the production of electronic components, leading to slight discrepancies in how individual devices detect, amplify, and process incoming IR signals. For example, small differences in the response times of photodiodes, the gain

of amplifiers, or the filtering characteristics of bandpass filters can cause variation in the time it takes for the receiver to detect and demodulate an IR pulse.

These differences in analog behavior, though minor, are consistent for each device, due to the fixed nature of the physical components. This makes it possible to fingerprint devices based on their unique signal processing characteristics. By analyzing how different receivers process the same signal, we can identify subtle, device-specific effects resulting from manufacturing variations. These variations, therefore, provide a reliable and repeatable basis for device authentication.

4. Methods

In this section, we detail the innovative data collection facility we developed, which includes 20 units of a widely used IR receiver commonly found in IoT deployments. Additionally, we describe the comprehensive set of experiments conducted using this setup.

4.1. Data Collection

In order to ensure that IRIS fulfills its objectives, a large quantity of high-quality data needs to be collected. To do so, we developed a data collection facility capable of collecting transmissions and receptions with high resolution, from a large number of IR receivers. The facility allows for remote programmability and control. As depicted in Fig. 8, the facility comprises two main components, one designed for reception and one for transmission:

4.1.1. Data Reception Component

Our experimental environment consists of 20 units of the VS1838B, a high-sensitivity 38 kHz IR receiver commonly used in IoT devices. This three-pin module (VCC, GND, and Signal) is easily integrated into circuits and microcontroller platforms like Arduino or Raspberry Pi.

Our data collection facility was specifically engineered to enable high-resolution data collection from multiple IR receivers. At the core of this facility is a unique staircase-style setup, 3D-printed using an FDM (Fused Deposition Modeling) printer with PLA+ filament. This setup ensures precise positioning of each of the 20 IR receivers in the same location relative to the IR emitter, which is installed on a full-length motorized breadboard. This innovative design maintains uniform experimental conditions across all

Figure 6: VS1838B IR receiver.

receivers, maximizing the consistency and reliability of our data. Notably, this facility does not require any modifications to the tested receivers.

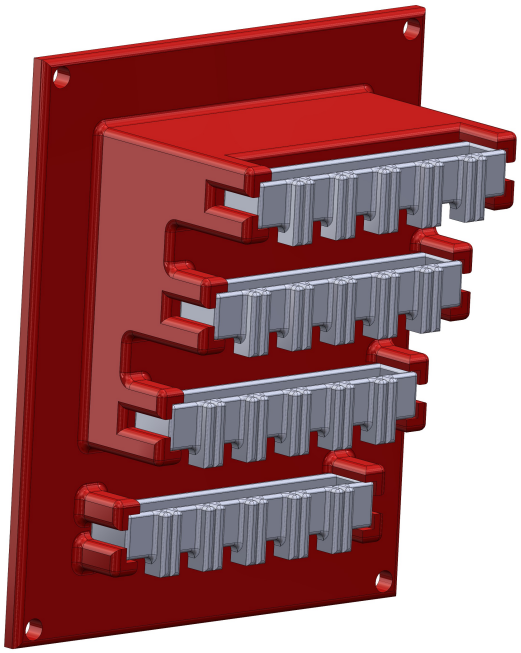


Figure 7: Design of IRIS's data reception component.

In order to collect high-quality transmissions and receptions, we used the PicoScope 6000 deep memory oscilloscope, configured at a sampling rate of 797 kS/s. One channel of the PicoScope 6000 was connected to the IR emitter, while the other channel was sequentially connected to each of the 20 IR receivers.

Because we aimed to examine how environmental variables impact our experimental results, we designed certain experiments to include variations in

temperature within the collection facility. Therefore, for each collected trace, we measured the instant temperature around the devices using a Raspberry Pi Sense Hat. Fig. 8 shows the data collection facility, which includes 20 IR receivers, which are connected to the ground, to a voltage source, as well as the motorized stage which connects each of them in turn to the PicoScope, while meeting the same conditions in front of an IR emitter. Adjacent to the setup is the Raspberry Pi Sense Hat used for measuring temperature. A brief video of the collection facility in action is provided in the link: `https://drive.google.com/file/d/1vHk-TZv3IfdXOAbf9D_22-pE16opWVrA/view?usp=sharing`.

4.1.2. Active IR Transmission Component

As described in 4.2, we evaluate our method in multiple environmental settings. For this purpose, we designed the collection facility to support two distinct settings: First, we conduct experiments in a *lab setting*, under controlled conditions, to identify phenomena that will inform the development of our method. Next, we evaluate the method's performance in practical scenarios using a *real-world environment setup*, implemented with a Samsung TV remote control as the transmitter.

In the lab setting, we placed an IR emitter on the full-length motorized breadboard. The emitter consists of an IR LED and an Arduino Nano Every. As for the real-world environment setup, we transmitted the signals from a common TV remote control, the Samsung BN59-01180A. To mimic the real-world use of a TV remote control as closely as possible, each transmission was sent from a slightly different location within the room, while we still used the motorized stage to ensure uniform conditions for the receiver under test.

In both settings, we transmitted the IR code for the digit '1' according to Samsung's IR protocol. To ensure that the transmission in the lab setting closely replicated the signal from the remote control, we first decoded the '1' button transmission from the TV remote, and then wrote a program that retransmitted the exact same code.

For each experiment, 100 traces were collected for each of the 20 receivers under consistent transmission settings, totaling 2000 traces per experiment. Each trace is approximately 3.9 MB in size and contains the receiver's identifier, 65000 records of timestamp, voltage (V) in Channel A (transmission), and voltage (V) in Channel B (reception). As mentioned earlier, to examine the impact of environmental variables on our results, one of the experiments

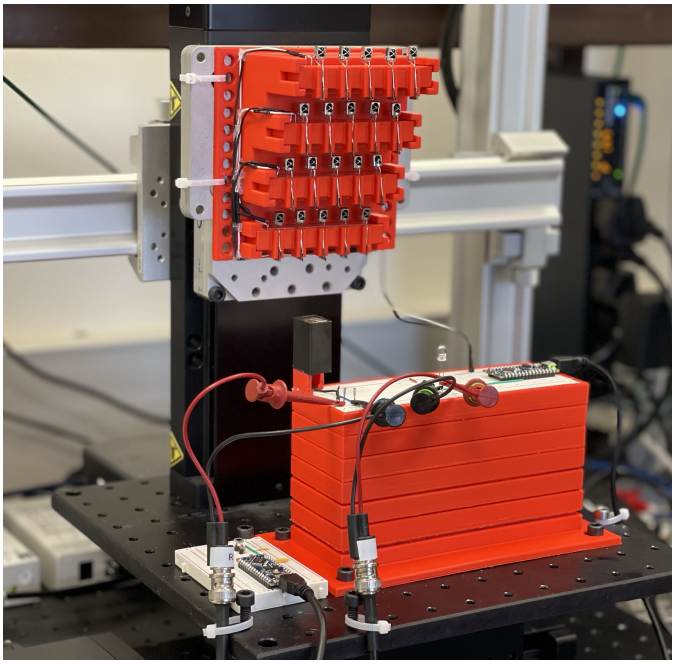


Figure 8: IRIS's data collection facility.

included the instant temperature as part of the trace.

4.2. Overview of Experiments

When considering the practicality of our proposed method, it is essential to evaluate its performance not only under ideal lab conditions, but also under realistic conditions that reflect challenges and constraints encountered in real-world IoT deployment settings. To this end, we evaluate our system under a variety of scenarios, as described below:

1. **Optimal Lab Conditions:** We evaluate the performance of a deep learning model under optimal lab conditions. Using full-length traces with both transmission and reception data at the full sampling rate, this setup allows us to assess the model's capabilities in ideal, controlled environments.
2. **Effect of Trace Length and Sampling Frequency:** We examine the impact of reducing the input trace length and sampling frequency on classification performance. In settings where computation and storage resources are limited, it may be necessary to reduce the trace length

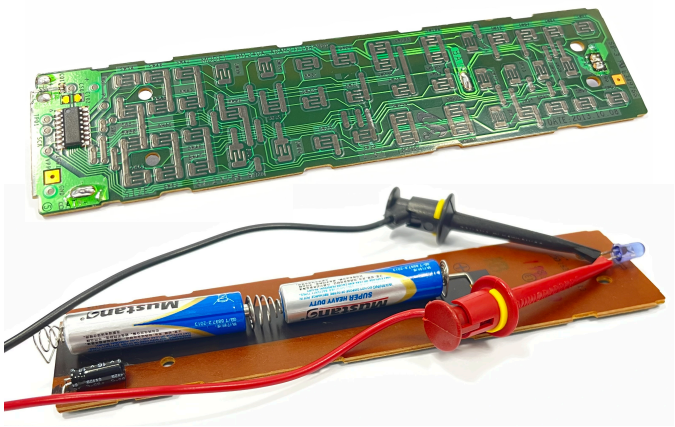


Figure 9: Dismantled Samsung BN59-01180A TV remote as part of the real-world environment setup.

and sampling frequency to conserve resources. To understand the implications of such reductions, we progressively shorten the trace length and simulate lower sampling frequencies by reducing data resolution. This investigation aims to identify an optimal trade-off that maintains high classification accuracy while accommodating resource-constrained environments.

3. **Receiver-only Identification:** Our ideal lab setting considers a case where the analog signal from both the transmitter and receiver is available. However, in some scenarios, only the receiver's characteristics may be accessible. To capture this setting, we evaluate the model's performance when only reception-related data is available, excluding the transmission data. This experiment allows us to investigate whether the receiver's characteristics alone are sufficient for robust classification and identification in such constrained settings.
4. **Real-World Environment:** We simulate a real-world authentication scenario by using a TV remote control as the transmitter. We evaluate the performance of the model in this uncontrolled environment to assess its robustness and adaptability when using everyday devices for authentication.
5. **Toward a Low-cost Solution:** We explore two potential approaches to reducing implementation costs and assess their impact on performance: first, by comparing the performance of classical machine learning algorithms with that of our deep learning model; second, by in-

vestigating the impact of using a low-cost Analog-to-Digital Converter (ADC) instead of an oscilloscope.

6. **Toward a Full Challenge-Response Protocol:** We move toward a more robust authentication method by sending multiple challenges to each receiver. On this experiment, the model is evaluated on its ability to distinguish not only between different receivers, but also between responses from the same receiver to different challenges, effectively demonstrating the potential of IRIS to implement a challenge-response protocol.

5. Results

The results of our experiments are summarized in Table 2. A detailed analysis of the results from each experiment is presented below.

Table 2: Summary of results

Setting	**Accuracy**	**Base Rate**
Optimal Lab Conditions	0.9855 ± 0.014	0.05
Reduced Trace Length	0.969 ± 0.01 (at 7.65 ms)	0.05
Reduced Sampling Frequency	0.978 ± 0.0166 (at 100 kS/s)	0.05
Receiver-only Identification	0.966 ± 0.0145	0.05
Real-World Environment	0.866 ± 0.02	0.05
Using Classical ML (Random Forest)	0.9995 ± 0.0015	0.05
Using Low-cost ADC	0.697 ± 0.0313	0.05
Challenge-Response (4 challenges)	0.9794 ± 0.0053	0.0125

*5.1. **Optimal Lab Conditions***

*5.1.1. **Evaluation of key performance metrics with a Convolutional Neural Network***

Our dataset consists of 2000 time-series traces divided into 20 distinct classes, with 100 traces per class. The length of each raw trace is 65000 samples. Across all deep learning-based experiments, we feed the raw time-series waveforms directly into the models without performing explicit feature extraction. Instead, the neural networks automatically learn meaningful representations from the transmission and reception signals. This enables the models to capture device-specific variations in IR receiver signal characteristics, which serve as the basis for authentication. In addition to the raw data,

we supplemented each trace with binary vectors generated by comparing each point in the transmission and reception channels to fixed thresholds.

We used a 1D Convolutional Neural Network (CNN) to classify the data. The CNN architecture consists of four convolutional layers with increasing feature maps, followed by a fully connected network. Regularization techniques included dropout ($p = 0.95$) and L2 weight decay (10^{-3}). The models were trained with the Adam optimizer (learning rate 5×10^{-5}, batch size 16) for up to 64 epochs, with early stopping applied if validation performance failed to improve for 11 consecutive epochs. Hyperparameters (learning rate, dropout ratio, batch size, and weight decay) were tuned empirically based on preliminary experiments under optimal lab conditions and then fixed across all CNN evaluations. All models were evaluated using 10-fold cross-validation, with each training fold further split to provide a validation set for early stopping. For each evaluation, key performance metrics—including average accuracy, standard deviation, median, minimum, and maximum accuracy—were recorded and analyzed.

The performance metrics demonstrate exceptionally high classification accuracy, with an average accuracy of 0.9855 and a low standard deviation of 0.014, significantly outperforming the base rate of 0.05. The results are consistent across all folds, with test accuracies ranging from 0.95 to 1.0. This consistent high accuracy demonstrates the model's effectiveness in this classification task.

As shown below, in non-ideal settings, the model was degraded by using a simpler machine learning pipeline or by providing less accurate data.

5.2. ***Effect of Trace Length and Sampling Frequency***

In real-world IoT applications, the ability to maintain high classification accuracy while working with limited data is essential. Constraints like storage limitations, processing power, and bandwidth can all lead to scenarios where the length of data traces or the available sampling rate must be reduced. Shorter trace lengths and lower sampling rates can both lead to a loss of critical information, potentially degrading performance. Therefore, understanding how these factors impact classification accuracy is crucial for optimizing system performance in resource-constrained environments. Here, we evaluate the effect of both trace length and sampling frequency to identify the minimum data requirements for achieving robust classification while maintaining high efficiency.

To evaluate the effect of trace length on the accuracy of our solution, we tested the same network on the same dataset multiple times, varying only the trace length—ranging from the full trace (81.5 ms) to the shortest possible trace (0), with intermediate lengths sampled at intervals of 0.12 ms. Any remaining part of shorter traces was padded with zeros to maintain consistency in input size.

To evaluate the impact of sampling rate on classification accuracy, we mimicked downsampling by systematically reducing the number of effective values in the traces while maintaining the original data length. This approach preserved the trace structure while progressively removing high-resolution details, effectively capturing the characteristics of reduced sampling rates.

As shown in Fig. 10, the results revealed a clear trend: for short traces (0-1.25 ms), the model achieved poor performance, likely due to insufficient temporal information. As trace length increased, the accuracy improved rapidly, reaching 0.65 ± 0.0527 at 5.26 ms and exceeding 0.95 beyond 6.14 ms. Notably, a local peak accuracy of 0.969 ± 0.01 was observed at 7.65 ms. This sharp increase in performance aligns with the fact that, on average, the transmission starts around 6.05 ms and the reception around 6.26 ms, providing the network with the key signal features needed to distinguish between classes. These timings are relative to our trace, as sampling began slightly before the broadcast was initiated. The model achieved its peak accuracy of 0.991 ± 0.0073 at 71.8 ms, and maintained strong performance at longer trace lengths. These findings indicate that, while longer traces significantly enhance the model's classification ability by providing more context, even trace lengths as short as approximately 6.14 ms suffice for achieving high accuracy in our task.

As also shown in Fig. 10, the classification accuracy exceeds 0.739 even before the first transmission. This suggests that the model identifies discriminative features associated with the IR receiver's passive behavior, including its response to environmental noise and baseline variations, which may inherently differ between devices.

The effect of lower sampling rates is illustrated in Fig. 11. As the figure shows, classification accuracy steadily decreases as the sample rate is reduced: At nearly 800 kS/s, the model achieves near-perfect accuracy, which drops only slightly at 400 kS/s and 200 kS/s (0.99 and 0.9875, respectively). However, below 100 kS/s, the impact becomes more significant, with accuracy declining to 0.98 at 100 kS/s and 0.965 at 50 kS/s, with sharper drops beyond that point.

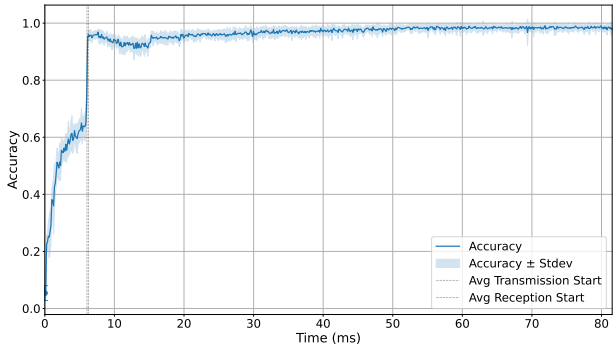


Figure 10: Effect of the trace length on the classification accuracy.

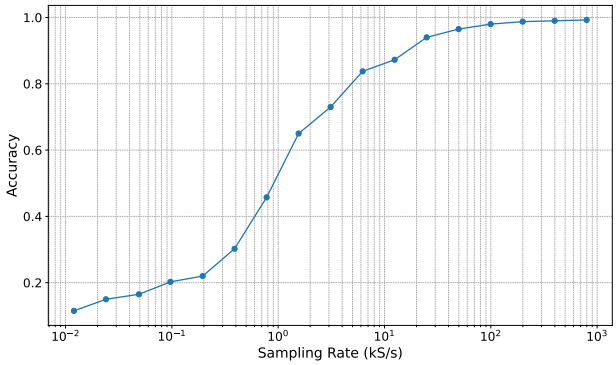


Figure 11: Effect of the sampling rate on the classification accuracy (logScale).

The evaluation of trace length and sampling frequency provides critical insights into optimizing the engineering and implementation of IRIS under constrained conditions.

1. **Trace Length:** Our findings show that longer trace lengths significantly improve model accuracy by providing richer temporal information. While short traces (less than 1.25 ms) yielded poor performance, accuracy improved rapidly as the trace length increased, reaching near optimal performance at approximately 6.14 ms, where accuracy exceeded 0.95. Beyond this point, the key signal features required for distinguishing between classes were fully captured, enabling the model to achieve peak accuracy. For applications with strict limitations on data storage or processing, using traces of at least 6.14 ms is recommended to balance performance and resource use.

2. **Sampling Frequency:** The analysis of reduced sampling rates indicates that the model can tolerate moderate reductions in data resolution without significant drops in accuracy. For real-world applications where bandwidth or energy efficiency is a concern, a sampling rate of at least 100 kS/s is recommended to maintain high accuracy. However, for systems where extreme resource limitations exist, sampling rates as low as 50 kS/s can still offer reasonable accuracy (around 0.965), providing a potential path for optimization in highly constrained environments.

5.3. ***Receiver-only Identification***

In many real-world applications, access to all available data channels, such as transmission and reception, may not always be feasible due to hardware limitations, bandwidth constraints, or system design. This raises the question of whether a model can still achieve high performance when only partial data, such as reception-related information, is available.

To evaluate the effect of using only the reception-related channels, we adjusted the optimal solution to include only the data relevant to reception. To achieve this, we excluded the transmission channel along with an additional derived channel that captures specific voltage characteristics.

The model maintained a high classification accuracy even without transmission data, achieving an average test accuracy of 0.966 ± 0.0145. This demonstrates that reception features alone are highly informative, enabling the model to effectively differentiate between the 20 classes. Such flexibility makes it suitable for systems where access to full data streams is impractical, offering a viable path to optimize performance while minimizing resource requirements. This approach is particularly useful in applications prioritizing reduced data complexity and cost.

5.4. ***Evaluating the Impact of Real-World Environments***

In real-world environments, several external factors can influence the behavior of infrared communication between transmitters and receivers. Interference from ambient light, reflections, and obstacles may distort signals; variations in transmitter battery level can alter signal strength; and differences in placement or distance between devices can increase variability. These effects are commonly highlighted in the literature as vulnerabilities in authentication schemes and are well recognized across different PUF modalities, making it difficult to maintain the same precision and reliability observed in controlled lab conditions [51, 52, 53].

To capture the impact of these practical conditions, we evaluated the method using the same architecture as in the optimal lab experiment, with an equivalent dataset obtained from a Samsung TV remote control serving as the transmitter. This setup provides a representative real-world environment in which to assess authentication performance. In line with real-world usage, transmissions were sent from slightly different locations within the room, while the motorized stage ensured that receiver placement remained uniform.

Despite the challenges associated with real-world scenarios, the model continues to perform robustly, achieving an average test accuracy of 0.866. Even with the inherent variability of real-world data, the model maintained a low standard deviation of 0.02, indicating consistent performance across different folds.

Clearly, the challenges posed by real-world settings, such as signal degradation and transmissions from varying locations, can significantly impact system performance. However, the results demonstrate that our model is robust and capable of generalizing to these conditions.

5.5. ***Toward a Low-cost Solution***

We further explored directions aimed at making the method more practical and resource-efficient. Two avenues were applied to this end: one focusing on the use of classical machine learning models, and the other on employing a low-cost ADC. To make the experiments comparable, we used the same architecture as the optimal solution to assess the model's performance.

5.5.1. ***Using Classical ML***

Driven by the motivation to provide a solution that performs consistently across different environments, we evaluated various Machine Learning models on our approach in both the lab setting and the real-world environment setup. This was achieved by extracting key features that characterize signal transitions in both transmission and reception, alongside statistical features that describe their variability. Specifically, we extracted transition-based features, including the timing of key voltage changes in the reception signal, relative transition times computed with respect to the first transmission event, and the duration of stable segments between consecutive transitions. Additionally, we derived processing time features, which measure the time delay between a transmission event and its corresponding response in the receiver. To further capture the device-specific signal behavior, we computed statistical

features for transition and processing times, including mean, median, standard deviation, variance, interquartile range (IQR), skewness, and kurtosis. This feature extraction process resulted in 732 features. We experimented with additional feature extraction approaches, including frequency-domain features (FFT-based) and alternative transition thresholds. However, these did not yield a meaningful improvement in classification performance. Thus, we proceeded with the 732 features, which provided the best balance between feature significance and computational efficiency.

As shown in Table 3, the accuracy of classical machine learning models approaches 1 in the lab setting, while achieving slightly lower accuracy in the real-world environment setup. These results demonstrate the potential of IRIS to function effectively even in computationally constrained scenarios where a deep learning approach may not be feasible. To illustrate this, Table 4 compares the inference speed of deep learning and classical machine learning models across different hardware platforms. The results show that in our setting, classical ML models achieve significantly higher inference speeds at comparable accuracies to deep learning models.

Table 3: Performance evaluation of machine learning models in the lab setting and in the real-world environment setup

Model	**Lab Setting**	**Real-World Environment Setup**
CatBoost	1.0	0.9659
LightGBM	0.9975	0.9682
Random Forest	1.0	0.9659
XGBoost	1.0	0.9591
Decision Tree	0.9825	0.9409
Gradient Boosting	0.9875	0.9682

Table 4: Inference performance of deep learning and classical ML on different hardware platforms (inf/sec)

Model	**GPU**	**Server**	**Raspberry Pi**
Deep Learning	660	28	Impractical
Classical ML	328	73	44

To further understand what contributes to the model's effectiveness, we analyzed feature importance scores using a Random Forest classifier with Mean Decrease in Impurity (MDI), also known as Gini Importance. This method evaluates the contribution of each feature by measuring how much

it decreases impurity (Gini impurity) when used for splitting in a decision tree-based model. The analysis reveals that processing time (represented by both values and indexes) and its statistical attributes play a dominant role in distinguishing between classes. Table 5 presents the top 20 most important features, highlighting the key statistical attributes that contribute to this distinction. Among the most influential features, various statistical measures such as the mean, median, quartiles, and sums of processing times appear frequently, indicating that both central tendency and distribution-based characteristics make a significant contribution to differentiation. Additionally, individual instances of processing time, particularly the first recorded instance, also contribute significantly, suggesting that the initial stage of processing contains important discriminative information. These empirical findings strongly validate the theoretical foundation of our work. The dominance of processing time in feature importance confirms that timing variations introduced by manufacturing tolerances in key components—such as the photodiode, amplifiers, bandpass filter, and comparator—provide a stable and repeatable basis for fingerprinting and authentication. This reinforces the idea that IR receivers, despite following the same design principles, exhibit device-specific timing behaviors that can be effectively leveraged for authentication.

To evaluate the tradeoff between the number of features and performance, we retrained Random Forest models using only the top-ranked features as determined by the MDI scores. Competitive accuracy was retained even with as few as five features (0.975), and with ten features the model already reached 0.984. Full accuracy was achieved with only 18 features, which is equivalent to using the entire 732-feature set. This confirms the feasibility of lightweight implementations of our method on resource-constrained platforms. Figure 12 illustrates the full tradeoff curve from 1 to 20 features, showing how accuracy increases steadily and converges to the full-feature performance.

Additionally, as demonstrated in Fig. 13, we evaluated the effect of trace length on classification accuracy, highlighting its impact on performance across different models, both with and without transmission data.

In the lab setting, where both transmission and reception data are available, the deep learning model exhibits remarkable adaptability to shorter trace lengths, achieving high accuracy rapidly. This performance is attributed to its ability to capture subtle discriminative features associated with the IR receiver's unique passive characteristics, such as environmental noise and device-specific baseline variations. All models converge to

Table 5: Top 20 most important features based on feature importance score

Rank	Feature
1	First Quartile Processing Time (Values)
2	Mean Processing Time (Indexes)
3	Sum Processing Time (Values)
4	Sum Processing Time (Indexes)
5	Median Processing Time (Values)
6	Third Quartile Processing Time (Indexes)
7	1st Instance of Processing Time (Values)
8	Minimum Processing Time (Values)
9	Third Quartile Processing Time (Values)
10	Median Processing Time (Indexes)
11	Mode Processing Time (Indexes)
12	Second Quartile Processing Time (Indexes)
13	Second Quartile Processing Time (Values)
14	Mode Processing Time (Values)
15	11th Instance of Processing Time (Values)
16	2nd Instance of Processing Time (Values)
17	Mean Processing Time (Values)
18	Median Post-Transmission Recovery Time (Values)
19	18th Instance of Processing Time (Values)
20	First Quartile Logical 0 Duration (Indexes)

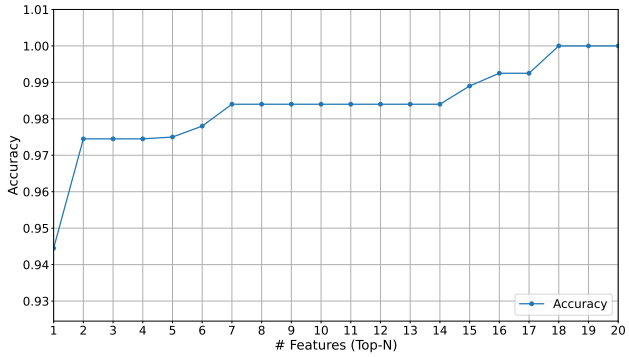


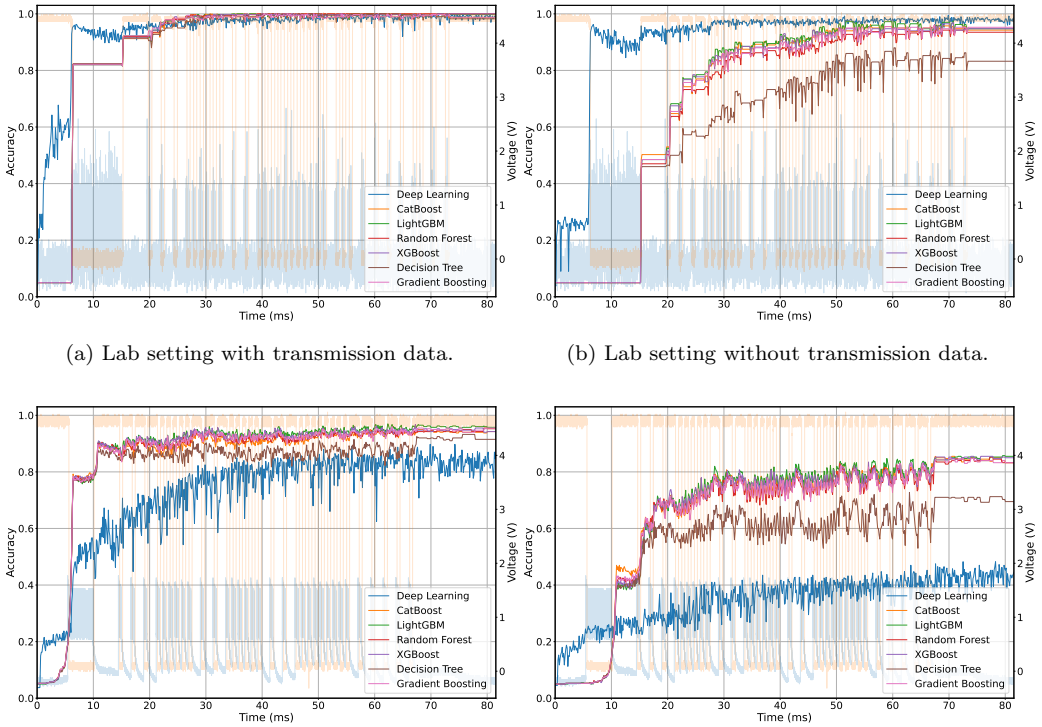
Figure 12: Accuracy as a function of the number of top-ranked features.

near-perfect accuracy for traces exceeding 30 ms, with minor variations attributable to noise or optimization nuances.

When transmission data is excluded in the lab setting, the deep learning model continues to excel, maintaining performance levels comparable to those

observed with transmission data. In contrast, classical machine learning models are significantly impacted, requiring substantially longer trace lengths to approach their optimal accuracy.

In the real-world environment setup, the deep learning model initially outperforms classical machine learning models at shorter trace lengths, leveraging its feature-capturing capabilities. However, as trace lengths increase to the mid-to-high range, classical machine learning models surpass deep learning. Notably, when transmission data is excluded from this setup, machine learning models exhibit lower peak performance, clustering around 0.85, and require longer traces to reach these levels. Conversely, the absence of transmission data has a pronounced impact on the deep learning model, causing it to underperform relative to the machine learning models in this scenario.



(a) Lab setting with transmission data.

(b) Lab setting without transmission data.

(c) Real-world environment setup with transmission data.

(d) Real-world environment setup without transmission data.

Figure 13: Effect of the trace length on the classification accuracy in different settings.

5.5.2. ***Using Low-cost ADC***

In the context of developing a low-cost solution, we evaluated our method using a low-cost ADC by replacing the PicoScope 6000 with an Arduino Nano Every to collect the transmission and reception data.

The model achieved an average test accuracy of 0.697 ± 0.0313, reflecting relatively stable performance across different folds. While this represents a decrease in accuracy compared to high-end equipment, the performance remains reasonable considering the low base rate. These results highlight the feasibility of using low-cost hardware for data collection in this task, offering a viable alternative for scenarios where cost is a significant constraint, albeit with some trade-offs in accuracy.

5.6. ***Toward a Full Challenge-Response Protocol***

To enhance the robustness of our authentication approach, we evaluated its potential as a PUF by exploring its suitability as a challenge-response protocol. Such a protocol operates in two phases: the enrollment phase, where various challenge-response pairs are assembled, and the authentication phase, as illustrated in Fig. 14. During authentication, the cloud server, acting as the authenticator, issues a challenge to the edge device and measures its response. The authentication phase involves the following steps:

1. At the beginning of the authentication process, the edge device initiates an authentication request to the cloud server, which includes its ID.
2. The cloud server randomly selects a challenge from the device-specific challenge-response pairs list and sends it to the device.
3. The end user performs the challenge, and the device then sends the response as an IR reception to the server.
4. The cloud server receives the response and passes it to a pre-trained model.
5. Based on the response, the model identifies both the challenge and the edge device to which the response corresponds. If the predicted device matches the provided ID and the predicted challenge aligns with the issued challenge, the device is approved. otherwise, it is rejected.

Adopting this protocol, we conducted an experiment to evaluate our method's performance under a challenge-response setup. Each receiver was presented with four unique challenges, resulting in a total of 80 classes (20

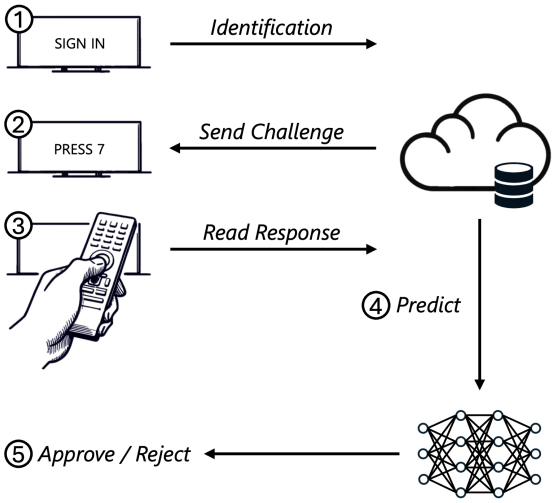


Figure 14: Illustration of the authentication process.

receivers, each with 4 unique challenges). The classifier was tasked with distinguishing not only between different receivers (e.g., Receiver 1 vs. Receiver 2) but also between the responses of the same receiver to different challenges (e.g., Receiver 1 + Challenge A vs. Receiver 1 + Challenge B).

The experiment achieved a classification accuracy of 0.9794 ± 0.0053, demonstrating the model's effectiveness in handling the challenge-response setup. These results strongly suggest that our method meets the criteria of a PUF, offering robust, device-specific authentication even under varied challenge conditions. Furthermore, the scalability of this approach is considerable; for example, a typical Samsung TV remote has 44 buttons, which could generate 44 distinct challenges in a real-world environment. In a controlled setting where the IR transmission can be customized further, the number of possible challenges becomes virtually limitless.

Overall, the experimental results demonstrate the robustness, adaptability, and scalability of our proposed method for secure, device-specific authentication. High accuracy was consistently achieved across optimal, constrained, and real-world conditions, underscoring the model's versatility. Key insights include the importance of trace length and sampling rate for optimal performance, the feasibility of receiver-only identification, and resilience to real-world challenges. The successful implementation of a challenge-response protocol further establishes the approach as a viable PUF, offering a secure and scalable authentication solution for IoT applications. Collectively, these

findings affirm the method's practicality for diverse operational scenarios while highlighting its potential for cost-effective, resource-efficient deployments.

6. Discussion

In this section, we explore the broader implications of our findings, evaluate the impact of environmental factors, such as temperature, on the performance of our method, and showcase practical demonstrations, while also outlining its limitations and future directions for development.

6.1. Evaluation of Temperature Impact on IR Receivers

Environmental factors, such as temperature, can influence the physical characteristics of key IoT device components [54], potentially affecting their fingerprinting stability and authentication accuracy. This section aims to evaluate whether IR receivers exhibit temperature sensitivity, rather than optimizing the model for cross-temperature authentication. To highlight any potential temperature impact, we intentionally overfit the model, allowing us to observe temperature-induced variations more clearly. In real-world scenarios, natural regularization and additional training strategies could mitigate these effects. To achieve this, we trained our optimized model on a dataset collected under two different temperature conditions: a low-temperature cluster and a high-temperature cluster. As seen in Fig. 15, the low-temperature samples were collected at temperatures equal to or less than 20.5°C (as measured by the Raspberry Pi Sense Hat), while the high-temperature samples were collected at temperatures equal to or higher than 23.5°C. We tested the model's performance across four scenarios: training and validating on the low-temperature cluster and testing on the low-temperature cluster, training and validating on the low-temperature cluster and testing on the high-temperature cluster, training and validating on the high-temperature cluster and testing on the high-temperature cluster, and training and validating on the high-temperature cluster and testing on the low-temperature cluster. This approach allowed us to assess the model's performance in both same-temperature and cross-temperature scenarios.

As shown in Table 6, the model performed exceptionally well when trained and tested within the same temperature conditions, achieving high accuracy (0.9545 ± 0.0252 for the low-temperature cluster and 0.972 ± 0.0131 for the high-temperature cluster). However, when tested on data from a different

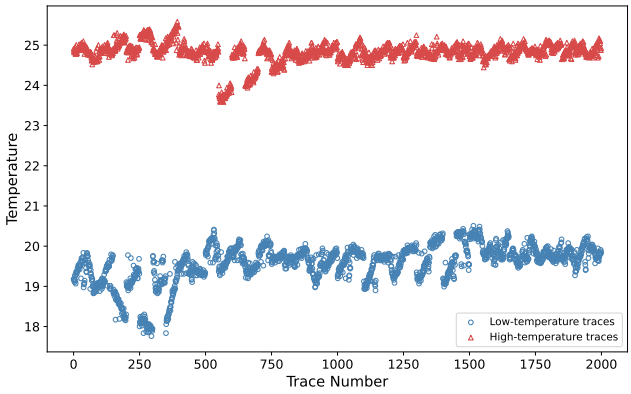


Figure 15: Low-temperature and high-temperature clusters.

temperature condition, the performance dropped significantly, with accuracies of 0.3013 ± 0.0199 (low to high temperature) and 0.3021 ± 0.05 (high to low temperature). This sharp decline underscores the impact of temperature on IR receiver characteristics, reinforcing its role in shaping their physical response to incoming signals.

Importantly, this experiment was designed to isolate the effect of temperature on IR receiver characteristics, rather than to assess the performance of a fully optimized model. In practice, incorporating regularization or temperature-aware calibration could significantly improve robustness in real-world scenarios.

To further assess whether temperature sensitivity fundamentally limits the effectiveness of our method, we trained the model on all collected samples across both temperature conditions and evaluated its performance on the full dataset. This approach yielded an average accuracy of 0.942 with a standard deviation of 0.0186, demonstrating that training on a diverse dataset significantly improves robustness to temperature variations. Additionally, we explored incorporating the instant temperature as a feature in the model. This approach resulted in a slightly higher accuracy of 0.957, with a standard deviation of 0.01. These results suggest that while temperature variations impact fingerprinting properties, explicitly modeling them as a feature can contribute to improved classification performance. However, the observed accuracy gain remains modest, indicating that robustness is primarily achieved through dataset diversity and training strategies that account for these variations.

Table 6: Model performance across different temperature conditions

Train \ Test	Low-temp	High-temp	Both
Low-temp	0.9545 ± 0.0252	0.3013 ± 0.0199	
High-temp	0.3021 ± 0.05	0.972 ± 0.0131	
Both			0.942 ± 0.0186

6.2. Demo

After evaluating the accuracy of our proposed method under various conditions, our next step was to demonstrate its practicality and feasibility through a real-world demonstration. The demonstration includes a graphical interface that visually represents the layout of the 20 receivers in our data collection facility. Each receiver is represented as a button on the interface, corresponding to its physical position in the setup. For the demo, we used a pretrained model to simulate the classification task in real-time. When a user selects a receiver by clicking on its corresponding button, the facility's system automatically positions an IR emitter in front of the selected receiver. The system then captures a new trace using the PicoScope 6000 and performs a live prediction to determine which receiver was selected.

We conducted the demo using the two major scenarios evaluated in our study: the lab setting and the real-world environment setup. In the lab setting, we transmitted a signal using the IR emitter connected to the Arduino Nano Every, as described in 4. For the real-world environment setup, we aimed a Samsung TV remote at the collection facility to simulate a real-world signal. This setup not only demonstrated the system's ability to correctly classify the receiver under varying signal conditions but also introduced our exploration of its suitability as a challenge-response protocol, showcasing the system's enhanced security capabilities. A brief demo video showcasing both scenarios is provided at the following link: `https://drive.google.com/file/d/1qk2K98-QJRqFf1UJlMN7mozxJDChDMVt/view?usp=sharing`.

6.3. Security Analysis

Since IRIS relies on the inherent physical variations of IR receivers to uniquely identify IoT devices, it introduces both design considerations and potential security aspects. While the method benefits from the uniqueness of IR receiver characteristics, its reliance on Line-of-Sight (LoS) communication is an inherent property of IR-based systems. Additionally, like any

authentication system, IRIS may be exposed to adversarial threats, such as signal replay, injection, and Denial-of-Service (DoS) attacks. This section first discusses the inherent LoS dependency of IR-based authentication, followed by an analysis of intentional attack vectors, their security implications, and possible mitigation strategies.

LoS Vulnerability: IRIS, like any IR-based system, relies on a clear line-of-sight between the transmitter and the receiver to function effectively. This is not a security vulnerability but an inherent characteristic of IR communication, which is also observed in common consumer electronics such as remote-controlled televisions, air conditioning units, and smart home devices. In practical deployments, LoS dependency means that obstacles between the IR transmitter and receiver may temporarily disrupt authentication attempts if they block or attenuate the signal. That said, since the authentication process occurs within the framework of the device's normal operation—where LoS is already required for standard functionality—IRIS does not introduce any additional constraints beyond those inherent to IR-based interactions.

Signal Jamming and DoS Attacks: Since IR receivers operate by detecting modulated IR signals within a specific wavelength range, they are susceptible to intentional interference from adversaries attempting to disrupt authentication. An attacker could attempt to flood the receiver with IR noise, preventing legitimate signals from being processed and effectively causing a denial of service. However, it is important to note that such an attack does not enable device impersonation—it only disrupts authentication, temporarily preventing access without granting unauthorized entry. Furthermore, jamming requires the attacker to have physical proximity to place a jamming device near the authenticating device, which is unlikely in most real-world scenarios, particularly in consumer electronics applications such as home TVs and smart home systems. To minimize the impact of IR jamming and DoS attempts, several mitigation strategies can be considered. Since IR interference is typically temporary and localized, a simple yet effective approach is to implement redundant authentication attempts, where the system automatically retries authentication after a short delay. This ensures that momentary disruptions—whether due to intentional jamming, unintentional interference, or environmental noise—do not lead to authentication failures. Additionally, in security-critical applications, multi-factor authentication (MFA) can provide a robust fallback mechanism. If persistent jamming is detected, the system can prompt an alternative authentication method to ensure continuity of secure access. By combining these approaches, IRIS can maintain

high availability and reliability, even in scenarios where intentional jamming attempts occur.

Signal Replay and Injection Attacks: Since IR receivers passively detect incoming IR signals, an attacker may attempt to bypass authentication by replaying previously recorded IR transmissions or injecting artificially generated signals. However, IRIS does not rely on the transmitted signal itself for authentication but rather on the unique physical response of the IR receiver to incoming IR signals. This fundamental distinction significantly mitigates the effectiveness of replay and injection attacks. For a replay attack to succeed, the attacker would need to perfectly replicate not just the IR signal but also the distinctive response characteristics of the receiver, which are shaped by hardware-level variations. Similarly, injection attacks, where an adversary generates custom IR signals in an attempt to mimic a legitimate device, would fail unless the injected signals could induce an identical hardware-dependent response. Even in scenarios where an attacker has access to a legitimate device and attempts to replicate its authentication response, they would still struggle to reproduce it accurately. The physical variations that define the receiver's fingerprint are device-specific and difficult to emulate, making it highly impractical for an attacker to generate an IR fingerprint that matches a legitimate device.

While the risk of signal replay and injection attacks is inherently low due to the hardware-dependent nature of IR receiver responses, additional safeguards can further enhance security in high-assurance environments. One such safeguard is challenge-response authentication, where the system varies the IR stimulus in a controlled manner, requiring the receiver to generate a response tied to an unpredictable challenge. This ensures that a previously captured response cannot be reused successfully in a replay attack. Additionally, temporal analysis can be used to detect anomalies in signal timing, such as unexpected delays, that may indicate an injection attempt. In scenarios where an additional layer of protection is required, multi-sensor fusion—combining the IR receiver's response with other sensor data—can further increase attack difficulty without introducing significant computational overhead. These countermeasures provide additional resilience for deployments in security-sensitive environments where an elevated threat model may apply.

6.4. Limitations

The IRIS system demonstrates strong potential for IoT device authentication, but certain challenges require attention. One such factor is the system's sensitivity to environmental conditions, particularly temperature variations. However, our findings indicate that training on diverse temperature conditions or explicitly modeling temperature as a feature can effectively mitigate this effect. While temperature variations influence fingerprinting properties, they do not inherently prevent reliable authentication when accounted for in the model design.

Another limitation arises from the fact that IR receivers are, in some cases, tightly integrated with their ADC, and the firmware controlling this ADC may not be reprogrammable. In such cases, adding a dedicated programmable ADC in parallel may be a more practical engineering solution than attempting to modify the existing ADC to incorporate fingerprinting capabilities. Addressing these aspects will enhance the feasibility of IRIS in real-world applications.

6.5. Future Work

To further enhance robustness in real-world deployments, future work may explore refinements in temperature-aware modeling and dataset diversity.

Additionally, scaling the Challenge-Response protocol presents an opportunity to strengthen security and transition the system toward a strong PUF design by increasing the complexity and diversity of challenges. These enhancements could solidify the system's security framework and expand its applicability to highly sensitive IoT environments.

In a broader context, future work may explore multi-sensor integration, combining data from IR receivers with additional sensors such as accelerometers, gyroscopes, and temperature sensors. This fusion could enhance the system's robustness by leveraging the unique fingerprinting properties of IR receivers alongside complementary data sources. Such an approach would increase security by incorporating multiple sensing modalities and improve resilience to environmental variability, making the system more robust against changing conditions and potential attacks.

6.6. Artifact Availability

To promote reproducibility, we provide an open-source artifact repository containing all the resources necessary to replicate and extend our work. The

repository includes the complete datasets and the code used for data collection and analysis across all experiments. Additionally, it features the full demo code and 3D design files for the printed data collection facility.

6.7. Conclusion

The unique security challenges of IoT edge devices make them difficult to authenticate. IRIS is a novel method for IoT authentication, which leverages the random variations introduced during the manufacturing processes of IR receivers. Our approach began with a comprehensive survey of components commonly used in IoT devices, which led to the identification of IR receivers as promising candidates for authentication. We subsequently developed a collection facility to capture and analyze the inherent variations in IR receivers, ensuring a replicable methodology. Through extensive data collection and analysis, we demonstrated that these variations can be effectively utilized to authenticate individual devices, highlighting the potential of IR receivers as a cost-effective and reliable security solution. Additionally, we provided an open-source artifact repository to facilitate future research and experimentation.

Our evaluation shows that IRIS offers a sustainable and low-cost security solution for IoT edge devices without requiring additional hardware on the authenticating device. Furthermore, we addressed the challenges of transitioning from a controlled lab environment to real-world scenarios, demonstrating the method's robustness under various constraints. These findings highlight the adaptability and scalability of IRIS, making it an attractive solution for authentication in consumer electronics and other everyday applications.

References

[1] L. O'Gorman, Comparing passwords, tokens, and biometrics for user authentication, Proc. IEEE 91 (12) (2003) 2019–2020. doi:10.1109/JPROC.2003.819605.

[2] A. A. S. AlQahtani, Z. El-Awadi, M. Min, A survey on user authentication factors, in: 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2021, pp. 323–328. doi:10.1109/IEMCON53756.2021.9623159.

[3] A. Kama, M. Amar, S. Gaaton, K. Wang, Y. Tu, Y. Oren, Juliet-PUF: Enhancing the security of IoT-based SRAM-PUFs using the remanence decay effect, IEEE Internet of Things J. 10 (14) (2023) 12715–12727. doi:10.1109/JIOT.2023.3253258.

[4] J. S. Yalli, M. H. Hasan, L. T. Jung, S. M. Al-Selwi, Authentication schemes for Internet of Things (IoT) networks: A systematic review and security assessment, Internet of Things 30 (2025) 101469. doi:10.1016/J.IOT.2024.101469.

[5] J. Oh, S. Yu, J. Lee, S. Son, M. Kim, Y. Park, A secure and lightweight authentication protocol for IoT-based smart homes, Sensors 21 (4) (2021) 1488. doi:10.3390/S21041488.

[6] W. K. Ahmed, R. S. Mohammed, Lightweight authentication methods in IoT: Survey, in: 2022 International Conference on Computer Science and Software Engineering (CSASE), 2022, pp. 241–246. doi:10.1109/CSASE51777.2022.9759798.

[7] M. El-hajj, A. Fadlallah, M. Chamoun, A. Serhrouchni, A survey of Internet of Things (IoT) authentication schemes, Sensors 19 (5) (2019) 1141. doi:10.3390/S19051141.

[8] M. Sain, Y. J. Kang, H. J. Lee, Survey on security in Internet of Things: State of the art and challenges, in: 2017 19th International Conference on Advanced Communication Technology (ICACT), 2017, pp. 699–704. doi:10.23919/ICACT.2017.7890183.

[9] Z. Huang, Q. Wang, A PUF-based unified identity verification framework for secure IoT hardware via device authentication, World Wide Web 23 (2) (2020) 1057–1088. doi:10.1007/S11280-019-00677-X.

[10] G. Vaidya, A. Nambi, T. Prabhakar, V. Kumar T, S. Sudhakara, IoT-ID: A novel device-specific identifier based on unique hardware fingerprints, in: 2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI), 2020, pp. 189–202. doi:10.1109/IoTDI49375.2020.00026.

[11] B. Chatterjee, D. Das, S. Maity, S. Sen, RF-PUF: Enhancing IoT security through authentication of wireless nodes using in-situ ma-

chine learning, IEEE Internet of Things Journal 6 (1) (2019) 388–398. doi:10.1109/JIOT.2018.2849324.

[12] C. Herder, M. M. Yu, F. Koushanfar, S. Devadas, Physical unclonable functions and applications: A tutorial, Proc. IEEE 102 (8) (2014) 1126–1141. doi:10.1109/JPROC.2014.2320516.

[13] A. Shamsoshoara, A. Korenda, F. Afghah, S. Zeadally, A survey on physical unclonable function (PUF)-based security solutions for Internet of Things, Comput. Networks 183 (2020) 107593. doi:10.1016/J.COMNET.2020.107593.

[14] F. Zerrouki, S. Ouchani, H. Bouarfa, T2S-MAKEP and T2T-MAKEP: A PUF-based mutual authentication and key exchange protocol for IoT devices, Internet of Things 24 (2023) 100953. doi:https://doi.org/10.1016/j.iot.2023.100953.

[15] A. Oun, M. Niamat, PUF-based authentication for the security of IoT devices, in: 2023 IEEE International Conference on Electro Information Technology (eIT), 2023, pp. 067–070. doi:10.1109/eIT57321.2023.10187363.

[16] T. Idriss, H. Idriss, M. Bayoumi, A PUF-based paradigm for IoT security, in: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), 2016, pp. 700–705. doi:10.1109/WF-IoT.2016.7845456.

[17] M. Cortez, A. Dargar, S. Hamdioui, G. J. Schrijen, Modeling SRAM start-up behavior for physical unclonable functions, in: 2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT 2012, Austin, TX, USA, October 3-5, 2012, IEEE Computer Society, 2012, pp. 1–6. doi:10.1109/DFT.2012.6378190.

[18] Y. Son, J. Noh, J. Choi, Y. Kim, GyrosFinger: Fingerprinting drones for location tracking based on the outputs of MEMS gyroscopes, ACM Transactions on Privacy and Security (TOPS) 21 (2) (feb 2018). doi:10.1145/3177751.

[19] M. Saideh, J. Jamont, L. Vercouter, Opportunistic sensor-based authentication factors in and for the Internet of Things, Sensors 24 (14) (2024) 4621. doi:10.3390/S24144621.

[20] Z. Ling, C. Gao, C. Sano, C. Toe, Z. Li, X. Fu, Stir: A smart and trustworthy IoT system interconnecting legacy IR devices, IEEE Internet of Things Journal 7 (5) (2020) 3958–3967. doi:10.1109/JIOT.2019.2963767.

[21] K. Huang, Y. Zhou, K. Zhang, J. Xu, J. Chen, D. Tang, K. Zhang, HOMESPY: The invisible sniffer of infrared remote control of smart TVs, in: 32nd USENIX Security Symposium (USENIX Security 23), 2023, pp. 4553–4570.

[22] M. Kim, T. Suh, Eavesdropping vulnerability and countermeasure in infrared communication for IoT devices, Sensors 21 (24) (2021) 8207. doi:10.3390/S21248207.

[23] W. Wang, A. D. Singh, U. Guin, A systematic bit selection method for robust SRAM PUFs, J. Electron. Test. 38 (3) (2022) 235–246. doi:10.1007/S10836-022-06006-X.

[24] F. B. Tarik, A. Famili, Y. Lao, J. D. Ryckman, Scalable and CMOS compatible silicon photonic physical unclonable functions for supply chain assurance, Scientific Reports 12 (1) (2022) 15653. doi:https://doi.org/10.1038/s41598-022-19796-z.

[25] J. Guajardo, S. S. Kumar, G. J. Schrijen, P. Tuyls, FPGA intrinsic PUFs and their use for IP protection, in: P. Paillier, I. Verbauwhede (Eds.), Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings, Vol. 4727 of Lecture Notes in Computer Science, Springer, 2007, pp. 63–80. doi:10.1007/978-3-540-74735-2_5.

[26] G. E. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in: Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, June 4-8, 2007, IEEE, 2007, pp. 9–14. doi:10.1145/1278480.1278484.

[27] J. Lee, D. Lim, B. Gassend, G. Suh, M. van Dijk, S. Devadas, A technique to build a secret key in integrated circuits for identification and authentication applications, in: 2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No.04CH37525), 2004, pp. 176–179. doi:10.1109/VLSIC.2004.1346548.

[28] S. Lee, M.-K. Oh, Y. Kang, D. Choi, RC PUF: A low-cost and an easy-to-design PUF for resource-constrained IoT devices, in: Information Security Applications: 20th International Conference, WISA 2019, Jeju Island, South Korea, August 21–24, 2019, Revised Selected Papers 20, Springer, 2020, pp. 275–285.

[29] C. Keller, F. Gürkaynak, H. Kaeslin, N. Felber, Dynamic memory-based physically unclonable function for the generation of unique identifiers and true random numbers, in: 2014 IEEE International Symposium on Circuits and Systems (ISCAS), 2014, pp. 2740–2743. doi:10.1109/ISCAS.2014.6865740.

[30] F. Tehranipoor, N. Karimian, W. Yan, J. A. Chandy, DRAM-based intrinsic physically unclonable functions for system-level security and authentication, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 25 (3) (2017) 1085–1097. doi:10.1109/TVLSI.2016.2606658.

[31] P. Prabhu, A. Akel, L. M. Grupp, W.-K. S. Yu, G. E. Suh, E. Kan, S. Swanson, Extracting device fingerprints from flash memory by exploiting physical variations, in: Trust and Trustworthy Computing: 4th International Conference, TRUST 2011, Pittsburgh, PA, USA, June 22-24, 2011. Proceedings 4, Springer, 2011, pp. 188–201. doi:10.1007/978-3-642-21599-5_14.

[32] S. Sakib, M. T. Rahman, A. Milenković, B. Ray, Flash memory based physical unclonable function, in: 2019 SoutheastCon, 2019, pp. 1–6. doi:10.1109/SoutheastCon42311.2019.9020567.

[33] A. M. Ali, E. Uzundurukan, A. Kara, Assessment of features and classifiers for bluetooth RF fingerprinting, IEEE Access 7 (2019) 50524–50535. doi:10.1109/ACCESS.2019.2911452.

[34] C. Labrado, H. Thapliyal, S. J. Prowell, P. T. Kuruganti, Use of thermistor temperature sensors for cyber-physical system security, Sensors 19 (18) (2019) 3905. doi:10.3390/S19183905.

[35] K. Rosenfeld, E. Gavas, R. Karri, Sensor physical unclonable functions, in: 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), 2010, pp. 112–117. doi:10.1109/HST.2010.5513103.

[36] S. Dey, N. Roy, W. Xu, R. R. Choudhury, S. Nelakuditi, AccelPrint: Imperfections of accelerometers make smartphones trackable, in: 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014, The Internet Society, 2014.

[37] D. Chen, N. Zhang, Z. Qin, X. Mao, Z. Qin, X. Shen, X.-y. Li, S2M: A lightweight acoustic fingerprints-based wireless device authentication protocol, IEEE Internet of Things Journal 4 (1) (2017) 88–100. doi:10.1109/JIOT.2016.2619679.

[38] Y. Lee, J. Li, Y. Kim, MicPrint: acoustic sensor fingerprinting for spoof-resistant mobile device authentication, in: H. V. Poor, Z. Han, D. Pompili, Z. Sun, M. Pan (Eds.), MobiQuitous 2019, Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Houston, Texas, USA, November 12-14, 2019, ACM, 2019, pp. 248–257. doi:10.1145/3360774.3360801.

[39] Y. Cao, L. Zhang, S. S. Zalivaka, C.-H. Chang, S. Chen, CMOS image sensor based physical unclonable function for coherent sensor-level authentication, IEEE Transactions on Circuits and Systems I: Regular Papers 62 (11) (2015) 2629–2640. doi:10.1109/TCSI.2015.2476318.

[40] R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, Physical one-way functions, Science 297 (5589) (2002) 2026–2030. arXiv:https://science.sciencemag.org/content/297/5589/2026.full.pdf, doi:10.1126/science.1074376.

[41] B. Gassend, D. E. Clarke, M. van Dijk, S. Devadas, Silicon physical random functions, in: V. Atluri (Ed.), Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002, ACM, 2002, pp. 148–160. doi:10.1145/586110.586132.

[42] D. E. Holcomb, W. P. Burleson, K. Fu, et al., Initial SRAM state as a fingerprint and source of true random numbers for RFID tags, in: Proceedings of the Conference on RFID Security, Vol. 7, 2007, p. 01.

[43] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, P. Tuyls, Extended abstract: The butterfly PUF protecting IP on every FPGA, in:

2008 IEEE International Workshop on Hardware-Oriented Security and Trust, 2008, pp. 67–70. doi:10.1109/HST.2008.4559053.

[44] Vishay Semiconductors, https://www.vishay.com/docs/80071/dataform.pdf.

[45] Texas Instruments, https://www.ti.com/lit/an/swra323/swra323.pdf?ts=1731137243803.

[46] Circuit Basics, https://www.circuitbasics.com/arduino-ir-remote-receiver-tutorial/.

[47] Tinkercad, https://www.tinkercad.com/.

[48] laskakit, https://www.laskakit.cz/user/related_files/vs1838b.pdf.

[49] Flexpcb, https://flexpcb.org/infrared-receiver-circuits-the-design-working-prin

[50] Vishay Semiconductors, Circuit description of IR receiver modules, https://www.vishay.com/docs/80069/circuit.pdf.

[51] A. Shamsoshoara, A. Korenda, F. Afghah, S. Zeadally, A survey on physical unclonable function (PUF)-based security solutions for Internet of Things, Computer Networks 183 (2020) 107593. doi:https://doi.org/10.1016/j.comnet.2020.107593.

[52] O. A. Ibrahim, S. Sciancalepore, R. Di Pietro, MAG-PUFs: Authenticating IoT devices via electromagnetic physical unclonable functions and deep learning, Computers & Security 143 (2024) 103905.

[53] S. Bavikatti Mallikarjun, M. Dixit, A. Weinand, H. Schotten, Survey on hardware-based physical layer authentication in next generation networks, 2024.

[54] L. Coppolino, S. D'Antonio, G. Mazzeo, L. Romano, A comprehensive survey of hardware-assisted security: From the edge to the cloud, Internet of Things 6 (2019) 100055. doi:https://doi.org/10.1016/j.iot.2019.100055.