Optical Token Transmission for Secure IoT Device Discovery and Control in Ubiquitous Environments

Viktor Matkovic
viktor.matkovic@uni-due.de
University of Duisburg-Essen
Distributed Systems
Duisburg, NRW, Germany

Malte Josten
malte.josten@uni-due.de
University of Duisburg-Essen
Distributed Systems
Duisburg, NRW, Germany

Torben Weis
torben.weis@uni-due.de
University of Duisburg-Essen
Distributed Systems
Duisburg, NRW, Germany

Abstract

This paper presents a camera-based optical transmission system and architecture of secure access to Internet of Things (IoT) devices in ubiquitous computing environments. Devices emit light tokens, referred to as local identifiers (LIDs), which are captured and decoded by a nearby camera. A LID initiates a local discovery process, where a bootstrapping service links the LID to an IoT device within a local network, allowing interaction without physical contact or prior pairing. The system supports device setup and control in ad-hoc scenarios using standard commodity hardware. By connecting a temporary physical signal to a network device identity, it enables access without continuous connectivity or complex setup. The light signal serves as a temporary authentication mechanism, while the camera verifies physical proximity and enables positional discovery of nearby devices. It provides a simple, secure way to interact briefly with nearby devices and is well suited for deployment in smart environments, augmented reality interfaces, and other ubiquitous computing contexts.

CCS Concepts

• **Networks** → **Naming and addressing**; • **Human-centered computing** → **Mobile devices**.

Keywords

Visible light communication; Proximity verification; Ubiquitous computing; Device discovery; Ad-hoc device setup; Commodity hardware

ACM Reference Format:
Viktor Matkovic, Malte Josten, and Torben Weis. 2025. Optical Token Transmission for Secure IoT Device Discovery and Control in Ubiquitous Environments. In *Companion of the the 2025 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion '25), October 12–16, 2025, Espoo, Finland*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3714394.3756299

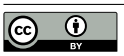
1 Introduction

Access to smart devices is often difficult for third-party users such as visitors, renters, or service personnel. These users typically do not have permanent credentials or prior pairing with the devices. In many cases, the methods for interacting with devices differ depending on the type and model. This results in inconsistent and time-consuming setup procedures. Some Internet of Things (IoT) devices are also constrained by automation routines or control layers that prevent direct interaction. For example, devices such as smart bulbs or smart displays may be embedded in predefined schedules or automation rules. These routines can block manual control or make it unclear how to temporarily override them. Existing access mechanisms, such as NFC or physical pairing buttons, require close physical proximity and direct interaction with the device. This is often not practical, especially when devices are mounted in hard-to-reach locations or managed remotely. There is a need for a general and low-effort mechanism to enable temporary access without requiring near-contact methods.

This paper presents a light-based data encoding and camera-based data decoding system and architecture for local discovery and control of IoT devices. Devices emit visible light signals that contain short-lived access local identifiers (LID). These signals can be captured by a camera-based decoding algorithm. These LIDs are then used to initiate a discovery process and enable device-level interaction. This method supports short-term, local control and can be applied to a wide range of device types in ubiquitous computing environments. The contribution of this paper is twofold. First, it describes the transmission method and the algorithm for encoding and decoding LIDs using visible light. Second, it introduces a bootstrapping architecture that enables device discovery and control through a local network service. This architecture allows users to interact with previously unknown devices using only visual contact and a standard camera interface. In addition, this system can enhance human-computer interaction through emerging augmented reality devices (e.g., Apple Vision). These devices can automatically detect and decode optical signals within the visual field. Discovered devices may then be represented in the user interface as virtual objects or avatars, accurately positioned at their physical locations. This enables users to interact with smart devices directly through the augmented reality environment, eliminating the need for physical interfaces or mobile applications.

2 Related Work

Visible Light Communication (VLC), also known as Li-Fi, typically uses modulated light from specialized LEDs or laser diodes and photodiodes for reception to achieve high data rates [1, 3]. Most VLC systems rely on dedicated hardware such as desktop lamps with integrated photodiodes and proprietary USB receivers, and are not compatible with built-in smartphone cameras.



UbiComp Companion '25, Espoo, Finland

ACM ISBN 979-8-4007-1477-1/2025/10
https://doi.org/10.1145/3714394.3756299

BlinkComm [4] demonstrates camera-based VLC using differential Manchester coding with microcontroller-controlled LEDs, achieving around 100 bps to photodiodes. While promising, it still requires tight emitter-receiver coordination and extremely close proximity, similar to NFC, limiting practical everyday use. In contrast, our method uses visible light signals emitted from commodity devices (such as displays or LED) and decodes them with existing smartphone or AR device cameras. This enables short-term, local control of previously unknown smart devices without network credential exchange, app installation, or specialized hardware.

The work by Haier et al. [2] explores the integration of Intelligent Reflecting Surfaces (IRS) into VLC systems to enhance (physical layer) security in indoor environments. It reviews existing methods, such as beamforming and interference management, and highlights the potential of IRS to improve both communication efficiency and security by dynamically controlling light reflection, which makes it undesirable for deployment in ubiquitous environments as it requires additional hardware and computational power.

The management system for hybrid VLC/RF systems, introduced by Trung et al. [5] focuses on smart retail applications. Although it demonstrates promising results in seamless handovers and user mobility tracking, it is not optimal for ubiquitous (IoT) systems, as it heavily relies on their hybrid structure and context-specific optimizations, thus lacking with regards to generalizability.

To our knowledge, no existing system combines camera-based VLC data transmission with a bootstrapping architecture that enables seamless, temporary access and device discovery in ubiquitous computing environments with commodity hardware. Furthermore, integrating this approach with AR platforms (e.g., Apple Vision Pro) allows users to discover and interact with virtual representations of devices directly in their visual field, providing a smooth, secure, and proximate control experience.

3 System Model

This section outlines the system model of our approach:

Proximity-Based Access: Device access is designed for users within a defined physical mid-sized range (1 to 5 meters), ensuring interaction occurs only in the immediate vicinity.

Camera Requirements: Users must have a device with a camera capable of detecting visible light signals (e.g., VR headsets, smartphones, webcams).

Limited Bandwidth: Our light-based data transmission supports only low data rates (camera frame rate is limited).

Discovery Latency Prioritization: To minimize IoT discovery latency, only small integers are transmitted (e.g., 8-bit or 16-bit values).

3.1 Threat Model

We assume that the goal of the attacker is to gain unauthorized access to an IoT device. We consider two modes of operation for our proposed architecture:

Non-Compromised Private Network: Any device or user with access to the private network is considered authorized and trusted to obtain the necessary information for pairing with a device. It is assumed that an attacker is able to observe and read data transmitted via the LED (e.g. using a camera or optical sensor). However, this information alone is insufficient to compromise the system, since it is assumed that the attacker does **not** have access to the private network.

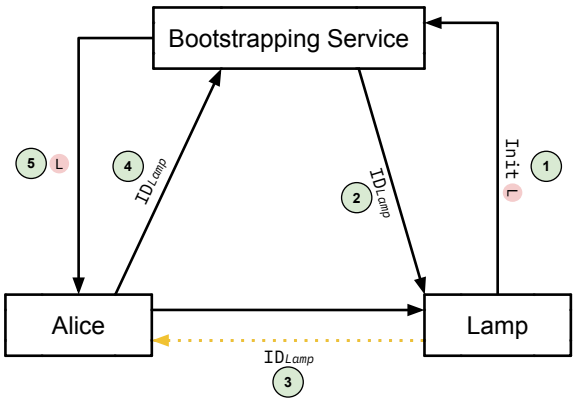


Figure 1: Architecture for non-compromised private networks where any device or user with network access is able to pair.

Compromised Private Network: The attacker has access to the network. It is assumed that the attacker may intercept and manipulate (spoof) communication within the network.

In Section 4, we present two architectures designed to address these threat models and mitigate potential risks under each mode of operation.

4 Architecture

The architecture is based on visual light communication to enable secure pairing with IoT devices within a private network (see Figure 1 and 2). It is based on light transmitted local identifiers (LID) which are small integers (e.g. 8-bit or 16-bit). It builds upon the system and threat models introduced earlier, addressing two modes of operation. The system consists of three base components: a light-emitting smart device, a receiving device (e.g., smartphone), and a local bootstrap service.

Non-Compromised Private Network. Figure 1 outlines the basic interaction process. First ①, a IoT device (Lamp) may initialize itself with the bootstrap service by providing the necessary information for pairing (e.g., URI, etc.). Next ②, the initialized IoT device is assigned a small random local identifier (LID). This LID is then continuously emitted via light signals for any receiving device equipped with a camera ③. The receiver (Alice) uses the detected LID to query the bootstrap service for the corresponding connection information ④ and ⑤. Finally, the receiver may establish a connection with the IoT device (Lamp).

Compromised Private Network. For the compromised network use case (Figure 2), we first introduce the necessary prerequisite information. The system assumes that only specific, authorized entities are allowed to use the service. This means that both users

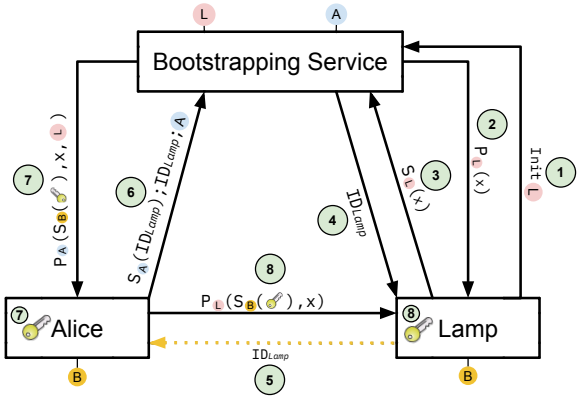


Figure 2: Architecture for compromised private networks where devices and users must gain prior authorization (even with network access) to be able to pair. In addition, the system is based on asymmetric cryptography.

and IoT devices must be trusted in advance. Trust can be established manually, through a user management system, or via a trusted signing authority. The system is based on asymmetric cryptography. The notation is as follows: $P_{\bigcirc}$ is the encryption function, using the public-key of the identity represented by $\bigcirc$. $S_{\bigcirc}$ is the signing function. The nonce x is chosen randomly, used for time-restriction purposes, and is unique for each IoT device. The Ⓑootstrap service (BS) is required to have the necessary information about all authorized users and devices for pairing (e.g., Ⓐlice and Ⓛamp). Alice and the Lamp do not need to know each other in advance; however, both must possess the public key Ⓑ of the BS. The initialization phase (① to ③) is a challenge–response used to register the Lamp and to limit the lifetime of the keys used in ⑦ and ⑧ by generating a nonce x. Furthermore, the nonce ensures that the LID is securely tied to the specific device, thereby preventing replay attacks. The initialization may be repeated to change the LID and nonce, thereby invalidating old keys. In ④ and ⑤, the LID is created (denoted as $\mathrm{ID}_{\mathrm{Lamp}}$), and the Lamp begins broadcasting the LID via light emissions. Alice receives the LID and sends it to the BS along with the signed LID by $S_{Ⓐ}$ and the user-identifying information Ⓐ in ⑥. The BS then verifies the received signature against the expected signature. Afterwards, in ⑦, the BS encrypts (using $P_{Ⓐ}$) the BS-signed key, along with the nonce x and the connection information of the Lamp Ⓛ. Alice now holds the key, which is later used for symmetric encryption between the Lamp and Alice. In ⑧, Alice then sends the same signed key along with the nonce x (encrypted using $P_{Ⓛ}$). The Lamp verifies the signature using the public key Ⓑ of the BS and checks the freshness of the nonce. If all verifications pass, both Alice and the Lamp share the same key for secure communication.

5 Light-Based Data Transmission

5.1 Encoding

According to Nyquist, the maximum bit rate at camera frame rate x is $\frac{x}{2}$, but with Manchester encoding doubling signal transitions, it is halved to $\frac{x}{4}$ to ensure correct decoding within frame rate limitations.

Table 1: The expected transmission lengths for different scenarios. Dividing the payload length n by the transmission length yields the respective overheads. Two exemplary cases for 8- and 16-bit payloads and a 5-bit pattern are also included.

Case	Transmission Length	8-bit	16-bit
Best-Case	$n + k$	62.5%	31.25%
Average-Case	$n + k + \frac{n}{2^k}$	65.63%	34.38%
Worst-Case	$n + k + \left\lfloor \frac{n}{k-1} \right\rfloor$	87.5%	56.25%

To identify the start of each data stream, we introduce a special k-bit pattern at the beginning of every transmission. Consequently, each transmission consists of n data bits plus the k-bit start pattern, resulting in a total length of $n + k$ bits. Assuming the data is uniformly random, we can derive the expected overhead for best, average, and worst case scenarios (see Table 1). In the best case, no additional stuffing bits are required because the payload does not contain the start pattern. However, since the probability of the k-bit start pattern naturally occurring in the payload is $\frac{1}{2^k}$, on average one stuffing event is needed every 2^k data bits, where each event inserts one extra stuffing bit. Thus, for a payload of length n, the expected number of stuffed bits is $\frac{n}{2^k}$. In the worst case, if the payload consists entirely of sequences identical to the start pattern, a stuffing bit must be inserted after every $k - 1$ bits to prevent confusion with the actual start pattern, leading to a total of $\left\lfloor \frac{n}{k-1} \right\rfloor$ stuffing bits. To calculate the overhead introduced by our approach, the number of stuffed bits can simply be divided by the minimum transmission length of $n + k$.

5.2 Decoding

Video frames are used to decode data signals from visual light patterns. The implementation is based on OpenCV. Video is captured from a camera with fixed frame dimensions to ensure consistent sampling (arbitrary size at initialization). Each frame is converted to grayscale to enhance contrast and simplify processing. The absolute difference (AbsDiff) between consecutive frames (x_{i-1}, x_i) is computed to detect temporal changes. A threshold $u = 25$ is applied to isolate significant transitions. To isolate significant pixel intensity changes for our transitions, the following fixed binary threshold was applied to the absolute difference:

$$\mathrm{binary}(x, y) = \begin{cases} 255 & \text{, if AbsDiff}(x_{i-1}, x_i) \geq 25 \\ 0 & \text{, otherwise} \end{cases} \quad (1)$$

The result is a binarized image in which pixels with intensity differences exceeding the threshold are set to 255 (white), indicating motion or change, and all others are set to 0 (black). Contours are extracted using Canny Edge Detection[1]. Bounding rectangles are computed around each contour to identify regions of interest, which are then validated according to the constraints of Manchester encoding, including the expected number and sequence of transitions

[1] https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html

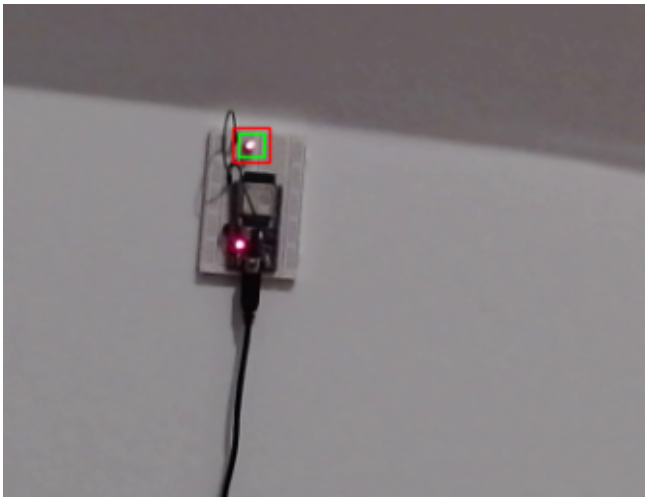


Figure 3: An experimental IoT device, emitting its LID. The red box is a bounding box, created by OpenCV, that encapsulates the detected light (by the receiver). The green box indicates a positive Manchester Code transition.

Algorithm 1 Decoding

Require: Video stream input
Initialize video capture device with specified resolution
Setup initial State configuration
while capturing video frames **do**
  Convert CurrentImage, PreviousImage to grayscale
  DiffImage = AbsDiff(CurrentImage, PreviousImage)
  Apply threshold and edge detection on DiffImage
  Extract contours from DiffImage
  for each contour in foundContours **do**
    Create transitionCandidate from contour
    Validate intersection with existing contour
    Determine transition based on pixel change
    Update timers and transition counters
    Validate transitionCandidate
  end for
end while

Validation is limited to contours that remain spatially stable across frames. Therefore, camera motion must be minimized to ensure reliable decoding. Unstable or shifting regions are excluded from analysis. Transition detection is performed by measuring median pixel changes within validated regions. The sequence of transitions is interpreted using Manchester encoding to reconstruct the transmitted bit-stream (see Algorithm 1).

6 Evaluation

We implemented a prototype of our light-based access system using Go and OpenCV. The prototype runs on a standard laptop equipped with a Logitech C920 webcam. The camera captures video at a fixed resolution of 800 × 600 pixels and a maximum frame rate of 24 frames per second (fps), which imposes a fundamental constraint on the data transmission rate. Given this frame rate and the use of Manchester encoding, the theoretical upper bound for reliable bit decoding is approximately 6 bits per second.

Our implementation (see Figure 3) uses visible light emitted from an off-the-shelf LED and modulated via software-controlled brightness changes on a display screen. The webcam detects these light transitions in real time, and OpenCV processes the frame differences to extract encoded data. While the current setup operates in a controlled environment (stable lighting, fixed distance, minimal camera motion), it demonstrates the feasibility of decoding dynamic light-based identifiers with commodity hardware.

Despite the hardware limitations, the system successfully decodes short-lived access codes suitable for initiating local discovery and access control workflows. T his shows that even under low frame rate constraints, light-based bootstrapping using commodity webcams is viable for temporary access use cases in ubiquitous computing settings.

7 Future Work

The current prototype runs on a computer using a webcam limited to 24 FPS. In the future, we plan to explore infrared (IR) light for transmission. IR is invisible to the human eye, which makes it less distracting and more practical in shared spaces. We also aim to move away from a desktop setup and bring the system to smartphones and wearable devices. This means dealing with different hardware limitations like power, frame rates, and processing speed.

Another direction is adding motion tracking. Camera movement (e.g. handheld or head-worn devices) can cause instability during decoding. Using built-in sensors like gyroscopes or existing tracking systems in AR/VR could help compensate for this and make the system more reliable during everyday use.

Other areas to improve include better handling of ambient light, trying different encoding formats, and making the system compatible with more types of devices. Furthermore, continuous scanning scenarios (e.g. AR headsets) may also require user consent, since bootstrapping may expose user data.

References

[1] Svilen Dimitrov and Harald Haas. 2015. *Principles of LED Light Communications: Towards Networked Li-Fi.* Cambridge University Press.
[2] Maaz Haider, Amena Ejaz Aziz, and Rashid Iqbal. 2025. Can IRS assist PLS for indoor VLC? *Physical Communication* 71 (Aug. 2025), 102703. doi:10.1016/j.phycom.2025.102703
[3] Parth H. Pathak, Xiaotao Feng, Pengfei Hu, and Prasant Mohapatra. 2015. Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges. *IEEE Communications Surveys & Tutorials* 17, 4 (2015), 2047–2077. doi:10.1109/COMST.2015.2476474
[4] Toni Perkovič, Tonko Kovačevič, and Mario Čagalj. 2018. BlinkComm: Initialization of IoT Devices Using Visible Light Communication. *Wireless Communications and Mobile Computing* 2018 (2018), 8523078. doi:10.1155/2018/8523078
[5] Kien Trung Ngo, Stefano Mangione, and Ilenia Tinnirello. 2023. A Novel Intelligent Management System Architecture for Hybrid VLC/RF Systems in Smart Retail Environment. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking.* ACM, Madrid Spain, 1–3. doi:10.1145/3570361.3615725