Cyber Security aboard Micro Aerial Vehicles: An OpenTitan-based Visual Communication Use Case

This paper has been accepted for pubblication in the IEEE ISCAS2023 conference.

Cyber Security aboard Micro Aerial Vehicles: An OpenTitan-based Visual Communication Use Case

Maicol Ciani*, Stefano Bonato†, Rafail Psiakis‡, Angelo Garofalo*, Luca Valente*,
Suresh Sugumar‡, Alessandro Giusti†, Davide Rossi*, Daniele Palossi†§
* Department of Electrical, Electronic and Information Engineering - University of Bologna, Italy
† Dalle Molle Institute for Artificial Intelligence - USI-SUPSI, Switzerland
‡ Secure Systems Research Center - TII, United Arab Emirates
§ Integrated Systems Laboratory - ETH Zürich, Switzerland
Contact author: maicol.ciani@unibo.it

***Abstract*—Autonomous Micro Aerial Vehicles (MAVs), with a form factor of 10 cm in diameter, are an emerging technology thanks to the broad applicability enabled by their onboard intelligence. However, these platforms are strongly limited in the onboard power envelope for processing, i.e., less than a few hundred mW, which confines the onboard processors to the class of simple microcontroller units (MCUs). These MCUs lack advanced security features opening the way to a wide range of cyber-security vulnerabilities, from the communication between agents of the same fleet to the onboard execution of malicious code. This work presents an open-source System-on-Chip (SoC) design that integrates a 64-bit Linux capable host processor accelerated by an 8-core 32-bit parallel programmable accelerator. The heterogeneous system architecture is coupled with a security enclave based on an open-source OpenTitan root of trust. To demonstrate our design, we propose a use case where OpenTitan detects a security breach on the SoC aboard the MAV and drives its exclusive GPIOs to start a LED-blinking routine. This procedure embodies an unconventional visual communication between two palm-sized MAVs: the receiver MAV classifies the sender's LED state (on or off) with an onboard convolutional neural network running on the parallel accelerator; then, it reconstructs a high-level message in 1.3 s, 2.3× faster than current commercial solutions.**

I. Introduction

Autonomous Micro Aerial Vehicles (MAVs) are progressively gaining importance thanks to their ubiquitous sensing capabilities. In the Internet of Things (IoT) ecosystem, nano-drones, i.e., palm-sized MAVs, can acquire and process information from different locations by flying where their presence is more important [1]. Therefore, they can exchange crucial data with fixed infrastructure or other drones, i.e., swarm operations. Their miniaturized form factor enables a wide range of applicability, for example, in narrow spaces [2] and human surroundings [3], but it limits the class of processors they can host aboard. This *i*) lower-bounds the computational/memory complexity of the algorithms that can run aboard and *ii*) forces the main drone's mission computer to simple microcontroller units (MCUs) that lack advanced cyber-security features.

In this emerging new era of connected and collaborating IoT devices/nano-drones, reliable security and privacy mechanisms are needed to protect assets and data collected or generated [4]. The security cornerstone of IoT devices is the Root of Trust (RoT), where critical assets are kept isolated and protected, the code executed is authenticated, and its integrity is verified [5]. Most modern IoT devices rely on hardware to ensure their RoT and therefore build the whole security stack on top of it, following the *chain of trust* principle [5]. Despite RoTs provide a solid hardware/software security foundation, there are several types of attacks potentially compromising the drones' operations, such as man-in-the-middle, denial of service, spoofing, jamming, rogue data injection, routing attack, etc. [6].

Current Commercial Off-The-Shelf (COTS) nano-drones platforms, such as the Bitcraze Crazyflie typically host low-power 32-bit MCUs such as the STM32F4 as main mission computer [2]. This class of MCUs provides sufficient computing power to guarantee basic functionalities such as low-level control loops, state estimation, and cryptographic encoding. Although they lack both a security enclave and RoT; therefore, they can not guarantee hardware isolation of code execution or support full-fledged operating systems capable of software isolation of different parts of the applications running on them. Similarly, more computationally-capable SoCs for nano-drones, such as the GWT GAP8 processors [7] available as a companion board for the Crazyflie nano-drone, i.e., AI-deck, still lacks RoT and security enclave able to take control of the whole system in case of attacks. In this work, we present an open-source SoC design of a mission computer for autonomous nano-drones, which includes silicon secure enclave and RoT by integrating the OpenTitan reference design [1].

The SoC is built around a 64-bit RISC-V CVA6 core featuring full Linux support and an 8-core cluster of 32-bit RISC-V cores acting as a software-programmable accelerator enabling vision-based tasks. Our work uses and enhances OpenTitan, the first collaborative open-source RISC-V-based silicon RoT, to support service request handling through an System Control and Management Interface (SCMI) [2] mailbox, master on the host domain, and secure GPIO handling connected to LEDs.

[1] https://opentitan.org/
[2] https://developer.arm.com/documentation/den0056/d/?lang=en

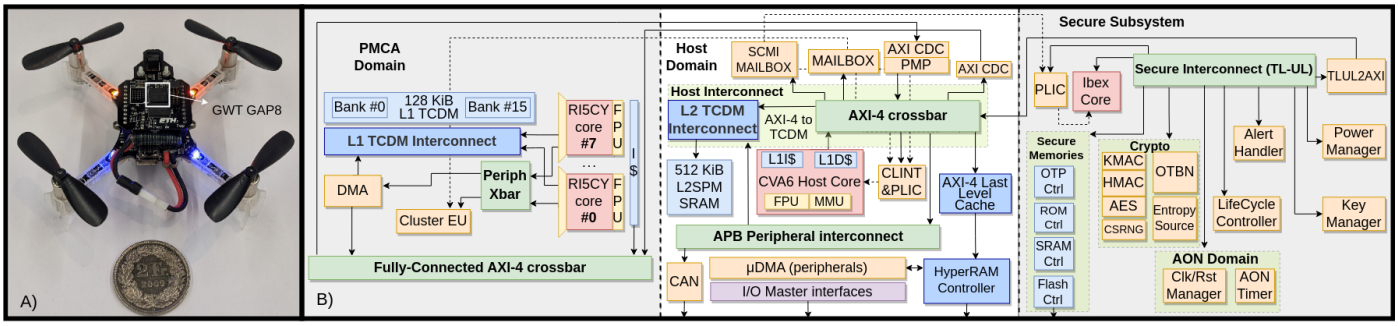

Fig. 1. A) The Bitcraze Crazyflie equipped with the GWT GAP8 SoC. B) The proposed SoC architecture envisioned as alternative MCU aboard the nano-drone.

We showcase our system design with a novel and field-proven use case of *Unconventional Visual Communication* (UVC) between nano-drones exchanging messages by LED blinking. In the context of visible light communication, machine learning techniques are used on the receiving end to implement signal demodulation [8], to recover the modulated signal from rolling-shutter images [9], [10], or to find locations of transmitters in images, before analyzing those regions of interest separately [11]–[13]. In contrast, our approach feeds raw images directly to a CNN to directly extract binary signal information (LEDs on or off), independently of the relative location of the drone transmitting the signal.

Once a cyber-attack compromises a nano-drone in the fleet, the radio channel cannot be trusted and the UVC is triggered, which depends exclusively on the secure OpenTitan sub-module and results in an SOS message emitted by the blinking LEDs. We show how other nano-drones equipped with the same SoC can reconstruct the SOS message by analyzing a video stream. By running a convolutional neural network (CNN), we assess the LED state of each input image. Then, a simple state machine continuously analyzes the series produced by the CNN and retrieve any custom message, such as the SOS one.

Our main contribution is the development of a novel SoC for drones' autonomous navigation providing cyber-security features which are keys in the proposed UVC-based use case. In detail: *i*) we integrate the OpenTitan secure subsystem into the navigation controller SoC; *ii*) we develop and field-test a simple light-based communication between multiple nano-drones in the swarm. With a power envelope of 250 mW and a silicon footprint of 9 mm^2, the proposed SoC can recognize an SOS message in 1.3 s performing 2.3× faster than a Crazyflie nano-drone equipped with an AI-deck, while offering support for a security enclave and full-fledged operating system.

II. System Architecture

This section presents the SoC architecture in Figure 1. It consists of a heterogeneous system architecture composed of a 64-bit application processor implementing flight control functions as well as auxiliary functions such as network stack, a parallel programmable accelerator for mission control functions, and a secure enclave based on OpenTitan IPs.

The SoC is built around the CVA6 core, a 6-stages, single-issue, in-order, 64-bit RISC-V core supporting the RV64GC ISA variant, virtual memory, three execution privilege levels, physical memory protection (PMP), and is capable of booting the Linux OS. CVA6 has 16KB of L1 I-cache and 32KB of write-through L1 D-cache, which enable simple coherency with other masters to the crossbar interconnect, which implements high-bandwidth, low-latency 64-bit AXI4 protocol. The *host domain* contains a scratchpad memory (L2SPM) and a complete set of peripherals such as I2C, (Q)SPI, CPI, SDIO, UART, CAN, PWM, I2S. Moreover, the host embeds also a standard Platform Level Interrupt Controller (PLIC), a Core Local Interrupt (CLINT), a controller for Cypress Semiconductor's external HyperRAM memories, and a Last Level Cache (LLC) to filter accesses to the external HyperRAM memory improving system performance. Peripheral data is transferred from/to the scratchpad memory through a dedicated DMA, called μDMA.

The Programmable Multi-Core Accelerator (PMCA) of the system is built around 8 CV32E-based processors which share 16×8KB SRAM banks (128KB L1SPM). The cores implement RV32 extension with many ML and DSP features such as hardware loops, MAC&Load operation, SIMD operations, and post-increment LD/ST. With SIMD, the operands' width can be reduced to double or quadruple the number of operations per cycle. The cluster also implements FPUs supporting FP32 and FP16 with SIMD support and features a two-level I-cache (512B for each core and 4KB shared) to speed-up execution of data-parallel tasks typical of drone mission control functions and deep neural networks for objects and pattern recognition. The architecture of the cluster is optimized for ML algorithms in embedded applications: it exploits scratchpad memories with DMA access, double buffering and custom ISA extensions to optimize memory utilization and computation.

The third key component of the drone navigation SoC is the secure enclave based on the OpenTitan architecture, acting as an on-chip Root of Trust (RoT), providing security services. The Ibex core is the main processor and is in charge to orchestrate the secure boot and all the RoT functionalities. Ibex controls four main components. The AON domain includes power, reset, clock management and a timer. The secure memories module includes One Time Programmable (OTP) memories which store security keys and seeds. The crypto module includes specialized accelerators such as Advanced Encryption Standard (AES), Hashing (HMAC and KMAC) and Big Number Accelerator (OTBN) for Rivest–Shamir–Adleman (RSA) and Elliptic Curve

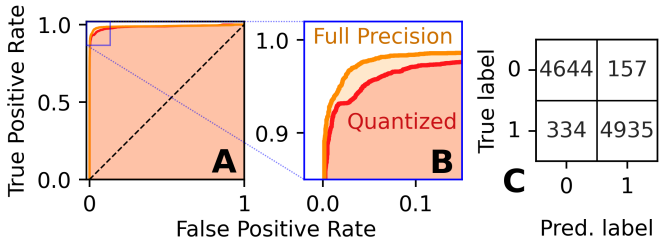

Fig. 2. Performance on the testing set. **A)** ROC Curves for full-precision and quantized models. **B)** Zoom-in of A. **C)** Confusion matrix.

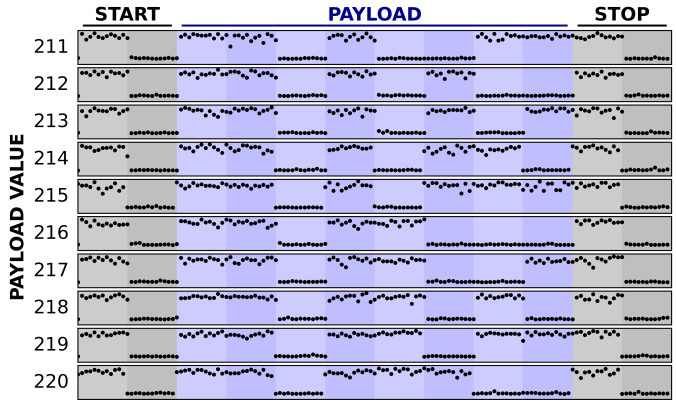

Fig. 3. Small dots denote individual CNN predictions of the transmitter LED state (12 per bit), before averaging. Vertical shaded areas denote the bit clock. Colors denote 2 start, 8 payload and 2 stop bits.

Cryptography (ECC). The security module includes key manager, life cycle controller and alert handler.

The host domain requires to exploit the hardware cryptography accelerators of the secure subsystem but it must not access directly its internal memory map for security reasons. Instead, it can only encode commands writing into a dedicated mailbox compliant with ARM System Control and Management Interface (SCMI). The host domain populates the shared memory of the mailbox and then it raises an interrupt to the secure subsystem's core through a dedicated memory mapped register. The Ibex core reads the content of the mailbox and executes the command encoded in it. At the end of the execution, the Ibex core raises back another interrupt to the host domain's core. Moreover, OpenTitan has its own internal timer that can trigger periodic interrupts. In this way the Ibex core can be periodically waken up in order to perform anomaly detection checks by analyzing the content of CVA6 and cluster's memories as well as external peripherals.

III. Security Use Case

Our use case envisions multiple nano-drones cooperatively operating and exchanging periodic data (e.g., mission commands, etc.) via radio (e.g., WiFi, BTLE, etc.). In this scenario, we address the following two *threat models*.

Man-In-The-Middle. A man-in-the-middle attack enables the attacker to intercept the communication and exchange malicious data with the drones. Following the zero trust policy [14], where we always authenticate and never trust, the drones periodically check that the received data are original and transmitted by an authenticated fleet member. If the authenticity cannot be verified, OpenTitan assumes that both the communication radio channel and the rest of the SoC are potentially compromised. Therefore, to notify the rest of the fleet about this situation, it enables the UVC blinking procedure by driving its secure GPIOs (exclusively connected to the secure subsystem) to transmit an informative SOS message. Other fleet drones – in line-of-sight with the transmitter one – can simultaneously or alternatively monitor peers' activity, distributing and time-interleaving the computational overhead for the message decoding.

Anomaly/Intrusion Detection. For this use case, we assume that there is a minimal anomaly/intrusion detection mechanism [15], [16] running on the Ibex core of OpenTitan, which is a secure region by construction. Since OpenTitan is the master of the TLUL-to-AXI interface on the host AXI-4 crossbar, it can monitor the activity of sensors (e.g., accelerometer, cameras, etc.). Then, if OpenTitan detects an anomaly, it can assume that the host domain, including the communication links, is compromised and triggers the UVC procedure. The specific implementation of a detection mechanism is out of the scope of this paper. Instead, we focus our work on the OpenTitan integration/isolation from the rest of the SoC and the implementation of the attack response method, i.e., the UVC exchanging SOS messages by demonstrating it with two Crazyflie nano-drones equipped with the AI-deck.

IV. Unconventional Visual Communication

CNN training. To decode the message transmitted by OpenTitan of a compromised drone, we use a lightweight CNN with the field-proven architecture of PULP-Frontnet [17]. The model input is a 160×96 pixels grayscale image; the output estimates the state of the LEDs of a Crazyflie nano-drone that is assumed to be visible in the image (either all on or all off), regardless of the drone position in the frame. Datasets are acquired with a monochrome QVGA camera from an *observer* drone facing a *transmitter* drone, which toggles its four LEDs every $\sim 0.4\,\mathrm{s}$.

The dataset is collected in a room equipped with a 18-camera Optitrack motion capture system, which tracks both drones. The transmitter drone flies in the central part of the room, while the observer records images while following a circular path around the transmitter. In this way, we maximize the variability of the images' backgrounds. The transmitter drone is automatically controlled in order to: *i*) always stay within the field of view of the observer; *ii*) move to random positions uniformly distributed on the observer's image plane; and *iii*) always lie at a distance of $0.2\,\mathrm{m}$ to $1.8\,\mathrm{m}$ from the observer. The dataset is composed of 36 flights, where frames are split to build a training set (the first 72%), a testing set (the intermediate 19%), and a validation set (the last 9%). A 10-frame gap is ignored between different sets in the same flight, to ensure that no similar frames appear in different sets. This results in $37\,\mathrm{k}$ frames for training, $10\,\mathrm{k}$

TABLE I
POWER CONSUMPTION AND AREA OCCUPATION

	Area (mm^2)	Leakage (mW)	Dynamic ($\frac{uW}{MHz}$)	Max Freq (MHz)	Max Power (mW)
Top	7.28	4.23	214.7	450	100.53
CVA6	0.49	4.79	47.5	900	47.54
PMCA	1.56	5.78	206	400	88.18
Mem Ctrl.	0.27	0.14	2.3	450	1.16
Opentitan	0.86	4.53	16	350	10.13
Total	7.28	19.47	486.5	-	247.54

frames for testing and 5 k frames for validation; each frame is labeled with the corresponding ground-truth LED state; the two states are equally represented in all sets.

Message encoding and decoding. Messages with an 8-bit payload are encoded to a simple self-clocking binary line protocol [18] that produces 12-bit packets, including 2 start bits and 2 stop bits. On our prototype, we employ the 8-cores GAP8 SoC manufactured in TSMC 55 nm technology capable of 22.65 GOp/s at 4.24 mW/GOp. With this SoC, the bit stream is transmitted by modulating the LED state at a rate of 2.5 bits per second; each 12-bit packet is therefore transmitted in 4.8 s. The observer drone acquires images at 30 frames per second (FPS). Each image is fed to the CNN, which estimates the LED's state in each frame. Each bit appears in 12 consecutive frames. First, the *bit clock* is determined from this sequence, then each bit in the bit stream is estimated by averaging the corresponding 12 CNN outputs and thresholding the result. Messages are decoded from the bit stream starting with a reserved start flag. If necessary, error detection and correction codes [19] can be implemented on top of this approach.

V. EXPERIMENTAL RESULTS

A. Deployment on Crazyflie with AI-deck

Figure 2 reports the CNN performance on the testing set; the full-precision model achieves an Area Under the ROC curve score of 98.88%, with negligible performance loss (-0.09%) after 8-bit integer quantization. After binarizing outputs at a threshold of 0.5, the model achieves an accuracy of 95.1%.

The end-to-end message transmission is assessed with an experiment in which the transmitter drone sends a sequence of 256 messages with a payload values from 0x00 to 0xFF. The observer drone, placed at a fixed distance of 30 cm, is always in line-of-sight with the transmitter one and decodes the received messages, at 30 FPS, with the quantized CNN. All 256 messages are decoded correctly. Figure 3 reports a subsequence of the received messages and a supplementary video demonstration is provided at https://youtu.be/TClcuUWJe0U.

B. Physical Implementation & Performance Evaluation

The proposed SoC has been implemented in the Global Foundries 22 nm FDX technology, employing the Synopsys Design Compiler for the logical synthesis and the place and route with Cadence Innovus. For the SoC's signoff we used the Synopsys PrimeTime, considering the worst case operating corner for a nominal supply voltage of 0.8 V (SS, 0.72 V, 125°C/-40°C), while power analysis was performed in typical operating conditions (TT, 0.8 V, 25°C) The layouts of the SoC and the main subsystems composing it are shown in Figure 4, while Table I summarizes the physical implementation.

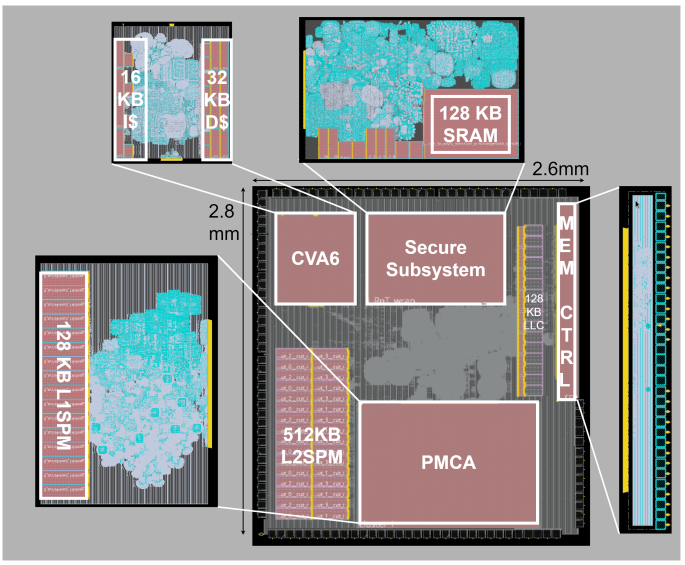

Fig. 4. In the middle, the layout of the SoC. Around it, the layouts of the PMCA, secure subsystem, HyperBus and CVA6.

To evaluate the performance of the proposed SoC on the UVC use-case, we first deploy our CNN on the PMCA of GAP8 SoC hosted on the COTS Carzyflie nano-drone. The full inference of the DNN on the proposed SoC takes 3.7 Mcycles, meaning that each payload's bit can be recognized by the receiver drone in 126 ms and that a full message described in Figure 3 can be recognized in 1.3 s by our proposed SoC. This is 2.3× faster than the same application running on the GAP8 SoC (@175 MHz). By coupling a linux-capable application core with a parallel programmable accelerator and a secure enclave within the power budget of 250 mW and a footprint of 9 mm^2, our SoC represents an appealing solution for secure and high-performance mission computers for nano-drones, paving the way for a wide range of new secure applications.

VI. CONCLUSION

We present an open-source SoC design for ultra-low power mission computers compatible with the limited power envelope of nano-UAVs. Our design provides sufficient computational resources to enable autonomous navigation tasks while enabling advanced hardware security such as RoT. The SoC is built around a 64-bit RISC-V CVA6 core featuring full support for Linux accelerated by an 8-core cluster of 32-bit RISC-V cores acting as a software-programmable accelerator for mission control tasks.. We integrate a security enclave based on an open-source RoT, enabling a multi-drone UVC use case. In our scenario, OpenTitan detects a security breach on the SoC and communicates an SOS message to a receiver drone by sending it through LEDs blinking. The receiver nano-drone can detect the message by running on the programmable accelerator a visual CNN and a simple state machine that decodes it. With

a power envelope of 250 mW and a silicon footprint of 9 mm^2, the proposed SoC can recognize an SOS message in 1.3 s performing 2.3× faster than a COTS baseline equipped with the GAP8 SoC, while offering support for a security enclave and full-fledged operating system.

REFERENCES

[1] S. A. Lakshman and D. Ebenezer, "Integration of internet of things and drones and its future applications," *Materials Today: Proceedings*, vol. 47, pp. 944–949, 2021.

[2] D. Palossi, J. Singh, M. Magno, and L. Benini, "Target following on nano-scale unmanned aerial vehicles," in *2017 7th IEEE international workshop on advances in sensors and interfaces (IWASI)*. IEEE, 2017, pp. 170–175.

[3] D. Palossi, A. Gomez, S. Draskovic, A. Marongiu, L. Thiele, and L. Benini, "Extending the lifetime of nano-blimps via dynamic motor control," *Journal of Signal Processing Systems*, vol. 91, no. 3, pp. 339–361, 2019.

[4] Y. H. Hwang, "Iot security & privacy: threats and challenges," in *Proceedings of the 1st ACM workshop on IoT privacy, trust, and security*, 2015, pp. 1–1.

[5] V. Kamakoti and S. Burman, "Building root of trust," in *2011 International Conference on Field-Programmable Technology*, 2011, pp. 1–1.

[6] M. Yahuza, M. Y. I. Idris, I. B. Ahmedy, A. W. A. Wahab, T. Nandy, N. M. Noor, and A. Bala, "Internet of drones security and privacy issues: Taxonomy and open challenges," *IEEE Access*, vol. 9, pp. 57 243–57 270, 2021.

[7] E. Flamand, D. Rossi, F. Conti, I. Loi, A. Pullini, F. Rotenberg, and L. Benini, "Gap-8: A risc-v soc for ai at the edge of the iot," in *2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, 2018, pp. 1–4.

[8] B. Lin, Q. Lai, Z. Ghassemlooy, and X. Tang, "A machine learning based signal demodulator in noma-vlc," *Journal of Lightwave Technology*, vol. 39, no. 10, pp. 3081–3087, 2021.

[9] C. Danakis, M. Afgani, G. Povey, I. Underwood, and H. Haas, "Using a cmos camera sensor for visible light communication," in *2012 IEEE Globecom Workshops*, 2012, pp. 1244–1248.

[10] K.-L. Hsu, Y.-C. Wu, Y.-C. Chuang, C.-W. Chow, Y. Liu, X.-L. Liao, K.-H. Lin, and Y.-Y. Chen, "Cmos camera based visible light communication (vlc) using grayscale value distribution and machine learning algorithm," *Opt. Express*, vol. 28, no. 2, pp. 2427–2432, Jan 2020.

[11] Y. Onodera, Y. Nakayama, H. Takano, and D. Hisano, "Drone positioning for visible light communication with drone-mounted led and camera," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2022, pp. 357–362.

[12] H. Takano, M. Nakahara, K. Suzuoki, Y. Nakayama, and D. Hisano, "300-meter long-range optical camera communication on rgb-led-equipped drone and object-detecting camera," *IEEE Access*, vol. 10, pp. 55 073–55 080, 2022.

[13] B. Chhaglani, A. S. Anand, N. Garg, and A. Ashok, "Evaluating led-camera communication for drones," in *Proceedings of the Workshop on Light Up the IoT*, ser. LIOT '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 18–23.

[14] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," National Institute of Standards and Technology, Tech. Rep., 2020.

[15] J. Galvan, A. Raja, Y. Li, and J. Yuan, "Sensor data-driven uav anomaly detection using deep learning approach," in *MILCOM 2021 - 2021 IEEE Military Communications Conference (MILCOM)*, 2021, pp. 589–594.

[16] W. T. Lunardi, M. A. Lopez, and J.-P. Giacalone, "Arcade: Adversarially regularized convolutional autoencoder for network anomaly detection," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2022.

[17] D. Palossi, N. Zimmerman, A. Burrello, F. Conti, H. Müller, L. M. Gambardella, L. Benini, A. Giusti, and J. Guzzi, "Fully onboard ai-powered human-drone pose estimation on ultralow-power autonomous flying nano-uavs," *IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 1913–1929, 2022.

[18] F. Halsall, *Data communications, computer networks and open systems*. Addison Wesley Longman Publishing Co., Inc., 1995.

[19] R. W. Hamming, "Error detecting and error correcting codes," *The Bell system technical journal*, vol. 29, no. 2, pp. 147–160, 1950.