Experimenting with Flash-based Out-of-Band Signaling between Android Devices

Alfred Anistoroaei
Faculty of Automatics and Computers
Politehnica University of Timisoara
Timisoara, Romania
alfred.anistoroaei@aut.upt.ro

Adriana Berdich
Faculty of Automatics and Computers
Politehnica University of Timisoara
Timisoara, Romania
adriana.berdich@aut.upt.ro

Bogdan Groza
Faculty of Automatics and Computers
Politehnica University of Timisoara
Timisoara, Romania
bogdan.groza@upt.ro

***Abstract*—Visible light communication between smartphones represents an interesting alternative for out-of-band communication and it may have security benefits as it doesn't rely on conventional channels that usually probed by adversaries. However, due to hardware and software limitations, achieving a solution that is simultaneously fast, reliable, and user-friendly remains challenging. In this paper, we design and implement a smartphone based communication system that addresses these challenges. The proposed solution uses the device flash LED for transmission and the camera in slow-motion mode for reception, with all processing performed on-device and without external hardware. The system is evaluated using Manchester and Non-Return-to-Zero (NRZ) encoding schemes across different distances and environments. Experimental results show that the proposed solution can achieve data transmission rates of up to 60 bps while maintaining a bit error rate as low as 3.00%, demonstrating a favorable trade-off between performance and usability compared to existing smartphone-only solutions.**

***Index Terms*—Visible Light Communication (VLC), Out-of-Band Channel, Security, Light Emitting Diode (LED), CMOS image sensor**

I. Introduction

Ensuring authentication and confidentiality in modern smartphone communications may pose challenges due to the globally accessible nature of conventional wireless channels. Indeed there are excellent solutions for it based on cryptography, but investigating additional channels is also of interest. In this context, visible light communication (VLC) represents a promising alternative, as it enables short-range, line-of-sight data exchange that inherently reduces the risk of interception and can support secure communication mechanisms. Previous research has explored various applications of VLC, including the transmission of raw data, cryptographic keys, localization information, and event triggers [1], [2], [3].

These works employ different approaches, such as using ambient light sensors [3], [2] or camera-based receivers [4], [5]. To achieve higher throughput, several studies exploit the rolling shutter effect of CMOS image sensors like [1], [6] or [7]. However, these methods typically require precise signal generation and external processing, limiting their practicality in everyday scenarios. While such approaches may achieve high data rates, they often involve increased system complexity or reduced usability, making them less suitable for general users. In particular, many solutions like [8], [7] rely on additional hardware such as Arbitrary Waveform Generators (AWGs) or offline processing using tools like MATLAB, e.g. [1] or [8].

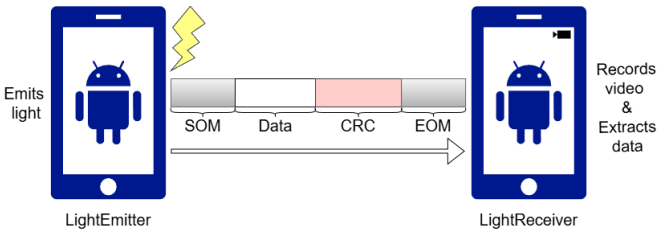


Fig. 1. System overview: suggestive depiction of two smartphones communicating via visible light channel from the smartphone's flash

In contrast, in this work we design and implement a smartphone-based VLC system that enables data transmission using only built-in device components. The proposed solution is therefore more usable in the sense that it does not require external devices, e.g., light sensors, LEDs and external processing. The entire communication stack is implemented directly on smartphones, from signal emission using the built-in LED flash to data reception through the camera in slow-motion mode, followed by on-device processing after recording, thereby allowing applications to operate seamlessly without the need for external devices, thus making them easier to use. Moreover, we take advantage of higher frame rates by using slow-motion recording, an option which is already available on many smartphones from the market. Thus the system can support increased data throughput without requiring major redesign efforts. Figure 1 provides a brief overview of the proposed solution. One smartphone acts as the emitter, transmitting a message encoded in visible light by flashing the LED of the smartphone, while a second smartphone acts as the receiver, recording the light signal and decoding the message. A frame is shown, beginning with a start-of-message (SOM), containing a data field, a CRC and an end-of-message field (EOM). The main advantage of the proposed solution is that we achieve a data rate of 60 bps, which places it between previous work that achieved only 5 bps with older devices and works that rely on external devices to boost performance and reach up to 10,320 bps. Our solution is developed in Android

Studio using Java, relies on Bytedeco libraries [9] and is tested on 2 smartphones.

The rest of the paper is organized as follows. In Section II we discuss related works. In Section III, we give a brief overview of the experimental setup and implementation. Section IV depicts the experimental results. Finally, Section V concludes our work.

II. Related work

In this section we present some related works on Visible Light Communication (VLC) and other out-of-band (OOB) channels.

A. Visible Light Communication Channels

In recent years, a considerable number of studies have investigated data transmission over OOB channels, with particular focus on VLC. In this section, we provide an overview of the most relevant works in this area and compare them with our proposed approach.

Among the reported solutions, the highest data throughput is typically achieved by systems exploiting the rolling shutter effect of Complementary Metal-Oxide-Semiconductor (CMOS) image sensors. This effect arises in sensors employing sequential row-wise readout and is widely available in commercial smartphones due to its lower cost and ease of implementation compared to global shutter devices. Representative works include [7], [8], [1], and [6]. In these studies, the optical signal is generated either using MATLAB or an Arbitrary Waveform Generator (AWG), driving an LED, while a smartphone camera captures the emitted signal. The recorded video is subsequently processed offline, typically using MATLAB-based image processing techniques.

From a security perspective, VLC offers several intrinsic advantages, including limited propagation range, resistance to electromagnetic interference, and improved user-assisted authentication. These properties make it particularly suitable for device pairing scenarios. For instance, [3] investigates the usability of pairing devices through OOB channels. Similarly, [2] employs both light and audio channels for secure transmission of cryptographic keys to sensor nodes, evaluating different encoding schemes such as Manchester and Pulse Width Modulation (PWM). In [10], the authors assess the feasibility of VLC under varying environmental conditions, ranging from low ambient light to direct sunlight, while also considering usability aspects. Another early contribution is [4], where each received video frame is reduced to a single scalar value by summing pixel intensities.

Beyond data transmission, visible light has also been employed in unconventional applications. For example, [11] demonstrates how variations in light intensity can be used to trigger malicious code on compromised devices, while [12] exploits imperceptible LED flickering for indoor localization. Specifically, [11] explores multiple triggering modalities, including light, audio, magnetic fields, and vibrations, showing that malware can be activated via LED lamps or display devices. In contrast, [12] enables localization by modulating LED lighting at frequencies invisible to the human eye, allowing devices to identify their position relative to ceiling-mounted light sources.

Table I summarizes and compares the data throughput achieved by representative works employing similar system configurations. There are only three papers, including the present work, that rely exclusively on smartphones. However, these studies achieve a lower throughput, i.e., 0.77 bps [3], and 5 bps [4], while our paper reaches up to 60 bps. Other works report significantly higher throughput like 1,680 bps [7] or 10,320 bps [1], but they rely on external hardware or offline processing techniques (e.g., MATLAB or AWGs), increasing system complexity and limiting their practicality. In contrast, the proposed solution enables a fully on-device implementation while maintaining a favorable trade-off between performance and usability.

B. Other Out-of-band Communication Channels

Another relevant point of comparison is with other out-of-band (OOB) communication channels, such as audio-based channels. Several works have investigated these alternatives. For instance, [13] provides a comprehensive survey of OOB communication techniques. Covert channels based on ultrasonic communication are analyzed in [14], while [15] explores the use of QR codes and NFC for IoT security applications. Hybrid approaches that combine multiple channels have also been proposed. In [16], both audio and visual channels are employed to exchange secret keys between Android devices. Specifically, one device transmits a secret key over the audio channel, while the other displays a QR code containing encrypted data along with a padded session key, following the SPEKE protocol [17]. While QR code transmission does not typically involve bit error rate (BER) evaluation, the audio channel exhibits significant variability. At short distances (e.g., 20 cm), the reported BER ranges from 0% to 9.36%, depending on the devices used. As the distance increases, the BER generally rises and remains inconsistent across measurements.

In contrast, the results obtained in this work using a visible light channel show a more consistent BER behavior across experiments. Although the achievable communication distance is shorter than in audio-based systems, the reduced variability may represent an advantage in terms of reliability and resistance to eavesdropping.

III. Implementation Details and Devices

In this section, we present the implementation details of the proposed system. We focus on the overall architecture and then discuss the transmitter and receiver components and the hardware employed during the experiments.

A. Overall application architecture

Modern smartphones are typically equipped with a camera for image and video capture, as well as an LED flash that can be used both for photography and illumination purposes. Furthermore, many recent devices support high-speed video recording in slow-motion mode, enabled by advanced image

TABLE I
COMPARISON WITH RELATED WORK IN TERMS BIT ERROR RATE (BER), THROUGHPUT, SETUP, NUMBER OF DEVICES AND PROCESSING PLATFORM

Paper	Throughput (bps)	BER	Setup	Devices	Processing
[3]	0.77	-	LED and ambient light sensor	2 Motorola Droid 1	On-device
[4]	5	-	LED or smartphone's display and camera	-	On-device
[2]	73	high error rate with timing and stability issues	LED and smartphone's display, development board and phototransistor	Arduino Uno and Android Smartphone	On-device
[10]	500	-	sensor nodes with LED and phototransistor	-	On-Device
[6]	896	0.38%	LED and AWG, Android smartphone	Cree XLamp XR-E, Tektronix, AFG 3252C, Samsung GTS7500	On-PC
[8]	900	10%-0.0001%	LED, smartphone and Matlab	Android smartphone and PC	On-PC
[7]	1,680	10%-0.001%	LED, smartphone	Smartphone and PC	On-PC
[1]	10,320	0%-0.2%	LED and AWG, smartphone and PC	Cree XR-E, Tektronix AFG3252C, iPhone 6, PC	On-PC
This work	**60**	**3.00%**	**Android smartphones**	**Motorola Moto e6Plus and Samsung Galaxy S6**	On-Device

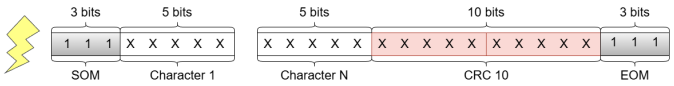


Fig. 2. Structure of the message, including 3-bit start and end delimiters, 5 bits allocated per character, and a 10-bit CRC field.

sensors and processing capabilities, reaching frame rates of up to 960 frames per second (fps) [18], [19].

Based on these capabilities, we developed two Android applications: *LightEmitter*, responsible for driving the LED to transmit data, and *LightReceiver*, which records the transmission in slow motion and processes the captured frames. Communication between the two applications is achieved through a custom protocol that works as follows. The start-of-message (SOM) and end-of-message (EOM) markers are encoded as three consecutive bits with logical value 1. Each data character consists of 5 bits, with the constraint that no character contains three consecutive logical 1, ensuring that it cannot be mistaken for SOM or EOM. Consequently, the protocol allows the transmission of $2^5 - 7 = 25$ distinct characters. A visual representation of the message structure is shown in Figure 2.

B. The Emitter and Receiver Modules

The *LightEmitter* application implements two encoding schemes, namely Manchester and Non-Return-to-Zero (NRZ), through dedicated classes. Upon initiating the transmission, a random message of 6 characters is generated and a 10-bit Cyclic Redundancy Check (CRC-10) [20] is appended. In total, 46 bits are transmitted: 30 bits payload, 10 bits for error detection, and 6 bits corresponding to SOM and EOM.

The use of SOM and EOM, together with the constraint that each character is encoded using 5 bits and that no character may contain three consecutive bits with a logical value of 1, enabled the application of a basic packet reconstruction technique. Specifically, the data-field length is a multiple of 5 bits. If three consecutive logical '1' values are detected within a character, then this indicates an error, implying that one bit (typically the last) is flipped. Although this approach does not guarantee correct packet reconstruction, it increases the likelihood of recovering valid data. Furthermore, SOM and EOM facilitate frame processing by triggering the application of the rolling shutter filter only after the SOM is detected and terminates the processing 10 video frames after the identification of the EOM. The 10 additional frames are intentionally processed after EOM detection since empirical observations showed that the EOM detection was occasionally wrong and more data was coming.

The bit stream is processed sequentially by the selected encoding scheme and transmitted by controlling the LED through the Android *CameraManager* API. For NRZ encoding, a delay of 14 ms is introduced between consecutive bits to satisfy the Nyquist–Shannon sampling criterion [21], taking into account the observed LED switching time of approximately 5 ms, resulting in an effective duration of 17 ms per bit. In the case of Manchester encoding, where transitions occur at twice the rate, the delay between transitions is reduced to 4 ms.

On the receiver side, video acquisition is performed using the default smartphone camera application in slow-motion mode. Due to platform constraints in Android that prevent simultaneous recording and real-time processing, the captured video is first stored and subsequently analyzed. The *LightReceiver* application processes the recorded frames by converting them to grayscale using Bytedeco libraries [9] and analyzing them sequentially. Brightness-based thresholding is used to differentiate between dark and illuminated frames, enabling detection of the SOM sequence. The flow of the two applications can be better observed in Figure 3.

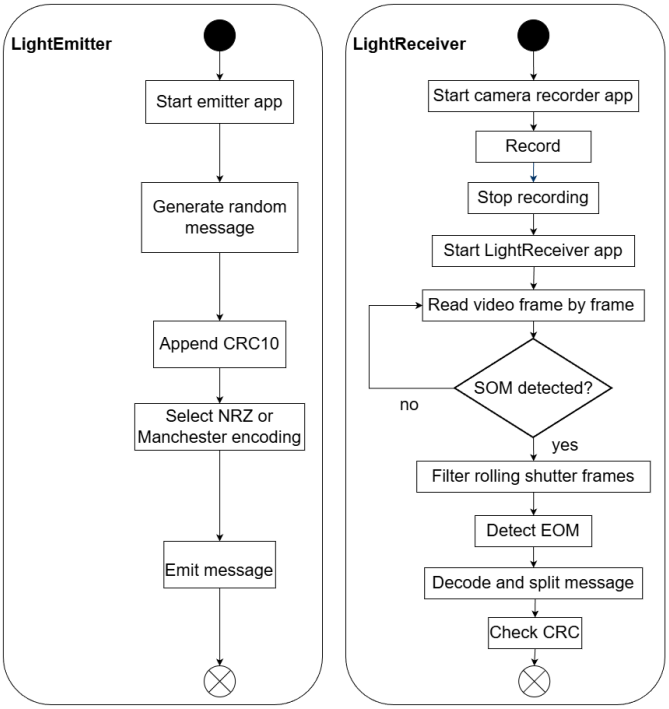


Fig. 3. Execution flow for LightEmitter (left) and LightReceiver (right) Android applications

The rolling shutter effect, inherent to CMOS sensors due to line-by-line image acquisition, introduces partially illuminated frames. To detect and filter such frames, an algorithm computes the mean pixel intensity for each row and identifies significant differences between adjacent rows. Frames affected by the rolling shutter effect are excluded from further processing. The remaining frames are then decoded according to the selected encoding scheme to reconstruct the transmitted message.

The experimental evaluation was conducted using multiple devices. Bit error rate (BER) measurements were obtained using a Motorola Moto E6 Plus running the *LightEmitter* application and a Samsung Galaxy S6 running the *LightReceiver* application. The BER is computed as the ratio between the number of incorrect bits and the total number of bits, not taking into account the SOM and EOM bits, as follows:

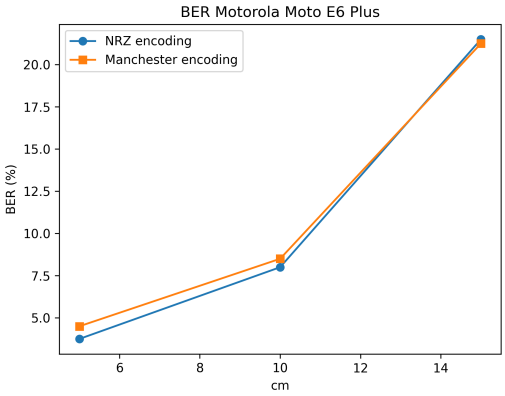
$$BER = \frac{N_{err}}{N_{tot}},$$

where N_{err} is the number of incorrect bits and N_{tot} is the total number of bits.

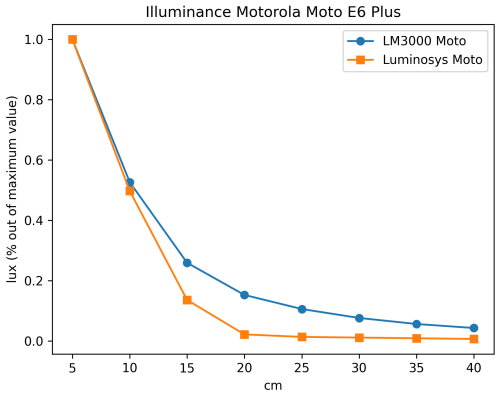
We calculated the value by sending 10 random messages of 6 characters. This means 40 bits for each message (1 character has 5 bits and CRC has 10 bits). In addition, illuminance measurements were performed using an iPhone 12, along with Samsung Galaxy A10, Galaxy J5, and Galaxy A3 devices.

IV. Experimental Results

In this section, we start from testing the device luminance and then present the experimental results for the two scenarios:



(i) Bit error rate evolution with distance for Motorola Moto E6Plus smartphone



(ii) Illuminance evolution with distance for Motorola Moto E6Plus smartphone

Fig. 4. Comparison of (i) BER and (ii) illuminance for Motorola Moto E6Plus smartphone

device in the glove-box of a car and indoors.

A. Testing the device luminance

To analyze the impact of distance on communication performance, we conducted additional measurements of the illuminance emitted by smartphone LEDs. As expected, illuminance decreases significantly with increasing distance, leading to a reduction in the signal-to-noise ratio at the receiver. This directly impacts the ability to distinguish between logical 0 and 1. This analysis explains the observed increase in BER with distance. Measurements were performed using an iPhone 12 alongside two light meter applications, namely [23] and [22]. These applications were calibrated to be used with a diffuser, for which we used a piece of white paper, as recommended by the aforementioned apps. In order to have results as precise as possible, we had to place the diffuser in front of the camera. Two applications were used due to observed discrepancies in absolute illuminance values under identical conditions. However, all applications consistently produced the same ranking of LED brightness across devices. The results are presented in Table II.Figure 4 (ii) illustrates the decrease in illuminance with distance for the Motorola device used in the transmission experiments.

TABLE II
SMARTPHONE LEDS ILLUMINANCE (LUX) MEASURED AT DIFFERENT DISTANCES USING TWO LIGHT METER APPLICATIONS

App.	Device	5cm [lux]	10cm [lux]	15cm [lux]	20cm [lux]	25cm [lux]	30cm [lux]	35cm [lux]	40cm [lux]
[22]	Motorola Moto E6Plus	1350	711	350	206	143	103	76	58
	Samsung Galaxy A10	3910	1110	516	304	194	150	118	93
	Samsung Galaxy J5	6990	2990	1520	850	595	405	287	246
	Samsung Galaxy A3	5970	1920	894	577	361	263	193	158
	Samsung Galaxy S6	2930	1390	625	373	248	193	132	103
[23]	Motorola Moto E6Plus	1330	662	181	29	18	15	12	9
	Samsung Galaxy A10	2488	1044	565	188	34	29	18	15
	Samsung Galaxy J5	8178	3283	1516	848	538	348	103	35
	Samsung Galaxy A3	4142	1816	815	491	226	41	29	23
	Samsung Galaxy S6	2172	1066	510	224	31	29	18	15



(i) Phones placed inside glove box (ii) Phones indoors at a distance of 10 centimeters

Fig. 5. The two phones placements in our experiments: (i) inside the car's glove box and (ii) indoors at small distance

TABLE III
BER FOR MANCHESTER AND NRZ ENCODINGS IN BOTH SCENARIOS, INDOOR AND GLOVE BOX

Environment	Manchester Encoding	NRZ Encoding
car - glove box	3.00%	3.25%
indoor - 5 cm	3.75%	4.50%
indoor - 10 cm	8.00%	8.50%
indoor - 15 cm	21.50%	21.25%

B. Results and discussion

In this subsection, we discuss the experimental evaluation of the proposed system, focusing on the achieved Bit Error Rate (BER) under different environmental conditions and distances. We also analyze the impact of encoding techniques and signal intensity on communication performance.

The experiments were conducted in two distinct environments in order to evaluate the impact of distinct light conditions. The first environment was the glove compartment of a vehicle, chosen due to its accessibility and almost complete absence of ambient light (when the lid is closed). The devices were positioned such that the transmitting LED and receiving camera were oriented towards the interior surface of the glove compartment, as in Figure 5 (i). The second experimental scenario was conducted indoors at night under minimal ambient lighting, ensuring reduced noise levels. In this setup, the devices were aligned such that the LED directly faced the receiving camera, as shown in Figure 5 (ii). Additionally, the distance between the devices was varied from 5 cm to 15 cm to evaluate the impact on the communication performance.

The measured Bit Error Rate (BER) for both Manchester and NRZ encoding schemes is presented in Table III. The results show that the lowest BER is achieved in controlled environments with minimal ambient light and short distances. In the glove compartment scenario, the BER is approximately 3.00% for Manchester encoding and 3.25% for NRZ encoding. Regarding indoor conditions, the BER increases with the distance, from 3.75% at 5 cm up to 21.50% at 15 cm. These results indicate that reliable communication is achievable at short distances (5–10 cm) and with as little noise as possible, where the BER remains below 10%. Beyond this range, the error rate increases significantly, limiting the usability. The illuminance measurements indicate that the flash of the Motorola device used in our experiments provides the lowest light intensity. Consequently, for applications requiring greater transmission distances, either a different device with a stronger flash can be employed or a standalone LED source can be used, as demonstrated in [6] and [1].

A comparison between Manchester and NRZ encoding shows that both schemes have similar performance across all tested scenarios. However, Manchester encoding consistently achieves slightly lower BER values. This can be attributed to its inherent self-synchronizing properties, which improve robustness against timing variations and frame misalignment. In contrast, NRZ encoding is more efficient in terms of symbol rate but is more sensitive under lower signal-to-noise conditions. Figure 4 (i) illustrates the BER for Manchester and NRZ encoding techniques for different distances, i.e., from 5 cm to 40 cm.

V. CONCLUSIONS

In this work, we demonstrated the feasibility of using visible light as a communication channel for data transmission between smartphone devices. The proposed system achieves a throughput of 60 bps over distances ranging from 5 cm to 10 cm under low ambient light conditions, with a measured bit error rate (BER) varying between 3.00% and 8.50%. The improved performance is also due to the use of the

slow-motion capturing capabilities of modern smartphones. Two encoding schemes, Manchester and Non-Return-to-Zero (NRZ), were implemented and evaluated. Both transmission and data processing were performed entirely on Android devices, without the need for external hardware or offline processing. The results obtained in this study indicate that even without exploiting the rolling shutter effect, it is possible to achieve data transmission rates beyond basic expectations using only the standard frame rate of smartphone cameras. As future work, we may extend the system by using smartphones capable of higher frame rates (e.g., 240 fps, 480 fps, or 960 fps), which are expected to significantly improve throughput and reliability. Another direction may be to investigate the use of different devices and external LEDs in order to evaluate the achievable transmission distances, as well as to examine system performance under varying ambient light conditions and daytime light.

ACKNOWLEDGMENT

This work was supported by RExQTCS: Romanian Excellence in Quantum Technologies enhancing Cybersecurity, a grant of the Ministry of Education and Research from Romania, CCCDI - UEFISCDI, project number PN-IV-P6-6.1-CoEx-2024-0214, within PNCDI IV.

REFERENCES

[1] W.-C. Wang, C.-W. Chow, C.-W. Chen, H.-C. Hsieh, and Y.-T. Chen, "Beacon jointed packet reconstruction scheme for mobile-phone based visible light communications using rolling shutter," *IEEE Photonics Journal*, vol. 9, no. 6, pp. 1–6, 2017, DOI: https://doi.org/10.1109/JPHOT.2017.2762460.

[2] J. Nilsson and J. Habbosh, "Out-of-band transfer with android to configure pre-shared secrets into sensor nodes," Master's thesis, Lund University, 2018.

[3] J. Han, Y.-H. Lin, A. Perrig, and F. Bai, "Mvsec: Secure and easy-to-use pairing of mobile devices with vehicles (cmu-cylab-14-006)," 2014, DOI: https://doi.org/10.1184/R1/6467987.

[4] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan, "Secure device pairing based on a visual channel," in *2006 IEEE Symposium on Security and Privacy (S&P'06)*. IEEE, 2006, pp. 6–pp, DOI: https://doi.org/10.1109/SP.2006.35.

[5] H. Okada, T. Hayashi, K. Kobayashi, T. Wada, C. B. Naila, and M. Katayama, "Visible light communication systems using a high-speed display and rolling-shutter camera," in *2023 28th Asia Pacific Conference on Communications (APCC)*. IEEE, 2023, pp. 244–249, DOI: https://doi.org/10.1109/APCC60132.2023.10460706.

[6] C.-W. Chow, C.-Y. Chen, and S.-H. Chen, "Enhancement of signal performance in led visible light communications using mobile phone camera," *IEEE Photonics Journal*, vol. 7, no. 5, pp. 1–7, 2015, DOI: https://doi.org/10.1109/JPHOT.2015.2476757.

[7] ——, "Visible light communication using mobile-phone camera with data rate higher than frame rate," *Optics express*, vol. 23, no. 20, pp. 26 080–26 085, 2015, DOI: https://doi.org/10.1364/OE.23.026080.

[8] Y. Liu, "Decoding mobile-phone image sensor rolling shutter effect for visible light communications," *Optical Engineering*, vol. 55, no. 1, pp. 016 103–016 103, 2016, DOI: https://doi.org/10.1117/1.OE.55.1.016103.

[9] https://bytedeco.org/, bytedeco libraries, accessed May 11, 2026.

[10] R. Roman and J. Lopez, "Keyled-transmitting sensitive data over out-of-band channels in wireless sensor networks," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. IEEE, 2008, pp. 796–801, DOI: https://doi.org/10.1109/MAHSS.2008.4660128.

[11] R. Hasan, N. Saxena, T. Haleviz, S. Zawoad, and D. Rinehart, "Sensing-enabled channels for hard-to-detect command and control of mobile devices," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, 2013, pp. 469–480, DOI: https://doi.org/10.1145/2484313.2484373.

[12] Q. D. Nguyen and N. H. Nguyen, "Mobile application for visible light communication systems: An approach for indoor positioning," in *Photonics*, vol. 11, no. 4. MDPI, 2024, p. 293, DOI: https://doi.org/10.3390/photonics11040293.

[13] B. Carrara and C. Adams, "Out-of-band covert channels—a survey," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, pp. 1–36, 2016, DOI: https://doi.org/10.1145/2938370.

[14] K. Block, S. Narain, and G. Noubir, "An autonomic and permissionless android covert channel," in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2017, pp. 184–194, DOI: https://doi.org/10.1145/3098243.3098250.

[15] S. Latvala, M. Sethi, and T. Aura, "Evaluation of out-of-band channels for iot security," *SN Computer Science*, vol. 1, no. 1, p. 18, 2020, DOI: https://doi.org/10.1007/s42979-019-0018-8.

[16] A. Anistoroaei, A. Berdich, P. Iosif, and B. Groza, "Secure audio-visual data exchange for android in-vehicle ecosystems," *Applied Sciences*, vol. 11, no. 19, p. 9276, 2021, DOI: https://doi.org/10.3390/app11199276.

[17] F. Hao and S. F. Shahandashti, "The SPEKE protocol revisited," in *International Conference on Research in Security Standardisation*. Springer, 2014, pp. 26–38, DOI: https://doi.org/10.1007/978-3-319-14054-4_2.

[18] Samsung, https://www.samsung.com/sg/support/mobile-devices/what-is-super-slow-mo-and-how-is-it-different-from-slow-motion-video/, samsung Super Slow Motion information page, accessed May 11, 2026.

[19] Sony, https://www.sony.co.uk/electronics/support/articles/SX577301, sony Super Slow Motion information page, accessed May 11, 2026.

[20] https://github.com/snksoft/java-crc, cRC-10/ATM parameters and implementation, accessed May 11, 2026.

[21] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949, DOI: https://doi.org/10.1109/JRPROC.1949.232969.

[22] L. I. GmbH, https://apps.apple.com/us/app/light-meter-lm-3000/id1554264761, iOS application, accessed May 11, 2026.

[23] M. Zwick, https://apps.apple.com/us/app/luminosys-light-meter/id6476405389, iOS application, accessed May 11, 2026.