Acta Polytechnica **65**(3):320–333, 2025
https://doi.org/10.14311/AP.2025.65.0320

Published by the Czech Technical University in Prague

MULTI-MODE SECURE DATA COMMUNICATION FOR REAL-TIME ROBOT ARM CONTROL WITH LI-FI TECHNOLOGY

Mohammed Majid Msallam[a,*], Refik Samet[b]

[a] *University of Technology, College of Control and Systems Engineering, Al-Wehda Neighborhood, 10001 Baghdad, Iraq*

[b] *Ankara University, Computer Engineering Department, Golbasi, 06000 Ankara, Turkey*

[*] corresponding author: 60190@uotechnology.edu.iq

Abstract. This work explores the secure data transmission approach to control a robotic arm using Li-Fi technology. Although Li-Fi offers built-in security by keeping the light within the walls of the environment, where it is used, leaks through windows are a concern. The proposed system addresses this concern by using a multi-mode approach with three options: without encryption, Rivest Cipher 4 (RC4) encryption, and Advanced Rivest Cipher 4 (ARC4) encryption. Users can choose the appropriate mode based on data sensitivity and processing constraints. The system transmits the movement commands to the robotic arm via a controller and transmitter. In used encryption modes, the received data is decrypted before instructing the robotic arm to perform the designated action. This secure Li-Fi system offers a promising solution for reliable and secure communication in industrial environments, addressing the limitations of traditional RF communication and improving data security.

Keywords: Secure communication, data encryption/decryption, robotic arm control, Li-Fi, RC4.

1. Introduction

The increasing number of wireless devices means a higher usage of radio frequency bandwidth. Therefore, Visible Light Communication (VLC) emerges as a solution. VLC leverages the vast bandwidth of the visible light spectrum, offering high data rates compared to base technologies that use radio waves such as Wireless Fidelity (Wi-Fi). However, Light Fidelity (Li-Fi) technology is an extension of VLC. The term Light Fidelity was formulated by German scientist Harald Haas in 2011 [1]. This technology uses Light Emitting Diode (LED) for data transmission, potentially alleviating bandwidth limitations and reducing energy consumption. LED bulbs that are used in Li-Fi technology are similar to those found in many energy-efficient offices and homes. Li-Fi technology includes a chip that modulates light for optical data transmission, making it a highly efficient and secure mode of communication. Data Transmission Architecture of Li-Fi technology is shown in Figure 1. Li-Fi technology uses LED for data transmission by turning its ON/OFF to encode data in binary form. Photodiodes convert received light signals into electrical signals. Li-Fi technology uses visible light to transmit data, enabling high-speed, bi-directional data transfer. Unlike radio waves, Li-Fi technology does not affect the health of humans [2]. This emerging technology leverages existing LED infrastructure for two purposes, illumination and data transmission, making it a promising solution for congested radio frequency spectrums. Wi-Fi technology and Li-Fi technology, although both technologies for wireless data transmission, use different spectrums. Wi-Fi technology uses the radio frequency spectrum, offering good range and compatibility with existing devices. However, radio waves are congested due to limited spectrum availability and can penetrate walls, increasing vulnerability to hacking. Conversely, Li-Fi technology uses the visible light spectrum, potentially enabling much faster data rates due to a wider bandwidth [3]. However, the range of Li-Fi technology is limited and confined to the area illuminated by visible light. Additionally, the inability of light to pass through walls enhances security within indoor environments. Table 1 summarises the difference between Wi-Fi and Li-Fi technologies.

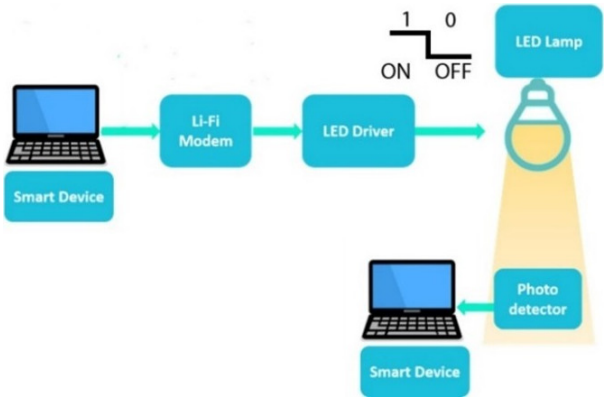


Figure 1. Data Transmission Architecture with Li-Fi technology.

The applications of Li-Fi technology extend beyond internet access. In healthcare, Li-Fi technology can securely transmit patient data within hospitals. In transportation, Li-Fi technology facilitates communication between vehicles, infrastructure, and pedestrians, using car headlights for data transmission. Li-Fi technology can also be used for indoor positioning

Parameters	Wireless Fidelity	Light Fidelity
Standard IEEE	802.11 a/b/g	802.15.7
Abbreviation	Wi-Fi	Li-Fi
Founder	National Cash Register Co	Harald Haas
Year	1990	2011
Speed Transfer Data	150 Mbps	224 Gbps
Communication medium	Radio waves	Light waves
Frequency	2.4 GHz	300 THz
Range	10 m to 100 m	10 m
Base station	Wireless access point	Li-Fi access point
Power Consumption	Medium	Low
Security	Less	High
Cost	High	Low
Compatibility	High	Low

TABLE 1. Comparison of Li-Fi technology and Wi-Fi technologies.

systems, creating location maps within buildings [4]. Furthermore, the high data rates of Li-Fi technology make it suitable for educational institutions, offering faster and more reliable connectivity compared to Wi-Fi technology. These various applications highlighted the broad potential of Li-Fi technology. In industrial automation, robotic arms play a crucial role in manipulating objects, handling hazardous materials, and performing tasks unsuitable for human intervention. These applications, such as loading/unloading toxic substances, conducting experiments in dangerous environments, or even performing surgery on patients exposed to radiation, necessitate reliable and high-speed data transmission. Li-Fi technology addresses this need by offering significantly faster data rates compared to traditional wireless solutions. Moreover, when operating in sensitive environments such as handling explosives, secure data transmission becomes paramount. The security features of Li-Fi technology combined with encryption methods such as those proposed by Umran et al. [5] can provide a robust communication channel for robotic arm control. Despite the security feature of Li-Fi technology, which is the limitation of the visible light reach, data vulnerability exists due to potential wall leakage through gaps or windows [6]. Data encryption is used to mitigate this risk. Encryption scrambles data into an unreadable format ensuring confidentiality and only authorised users with decryption keys can access the information. Encryption methods often involve exchanging secret keys beforehand. These keys are used for both encryption on the transmitter side and decryption on the receiver side, further safeguarding the data in transit. While encryption offers significant security advantages, it requires processing power, potentially impacting system speed and complexity. There are two main encryption categories: symmetric and asymmetric. Symmetric algorithms, such as the popular RC4, use a single shared key for both the encryption and decryption, offering speed and simplicity [7]. Conversely, asymmetric algorithms use public key pairs, improving the security during key exchange. The Diffie-Hellman key exchange is a prime example of asymmetric algorithms. By combining the security features of Li-Fi technology with encryption methods, the data transmission can be significantly secured. Li-Fi technology can experience packet loss due to various factors, such as obstructions, interference from ambient light, and mobility of devices, where packet loss refers to the failure of data packets to reach their intended destination [8, 9]. Overhead in Li-Fi technology arises from the inclusion of headers, control information, and error correction mechanisms within the transmitted data packets, impacting the overall delivered data. The energy cost of Li-Fi is primarily associated with the power consumption of the LEDs which are used for data transmission and are generally energy-efficient. This article does not address these aspects, as they have been discussed in other publications. This article addresses the security vulnerabilities arising from wall leakage in Li-Fi communication in real time. This work proposes a multi-mode approach offering flexibility in data transmission. The first mode transmits data unencrypted, while the second and third modes use encryption. The second mode uses the Rivest Cipher 4 (RC4) algorithm. The third mode uses the Advanced Rivest Cipher 4 (ARC4) algorithm, which was developed by Msallam [10]. ARC4 is specifically designed for fast implementation and high security on resource-constrained microcontrollers. However, although RC4 is easy to implement, does not need large storage space, and has less complexity, there are weaknesses in its output where the hacker can easily decrypt the data [11, 12]. Based on this reason, a third mode has been added that uses ARC4. Therefore, the novelty of this work is the design of a system that uses three modes to transmit data, where the user can choose one of them to send data according to the importance of the transmitted data. So, when the environment being used is safe or the data being transmitted is unimportant, the unencrypted mode is used. When the environment is dangerous or the data being

transmitted is important, RC4 mode is used. When the environment is very dangerous or the data being transmitted is sensitive, ARC4 mode is used. This paper aims to design and implement a robotic arm that is used to transport hazardous or toxic materials. The rest of this article is structured as follows. Section 2 presents the literature survey of works related to our work. Section 3 presents the general architecture of the proposed system. In Section 4, the used materials and methods are described, and the connecting material of the prototype is presented. This is followed in Section 5 by the implementation and the results of the proposed system. Finally, the conclusion and suggestions for future research are given at the end of this article.

2. Related work

Since the Li-Fi technology introduction in 2010, Li-Fi technology research has encompassed various design and implementation aspects. Fuada et al. presented a Li-Fi system using a ZYBO Zynq-7000 microcontroller [13]. Their transceiver used an Analog Front End (AFE) with a photodiode receiver and a High Brightness LED (HBLED) transmitter for downlink communication, and an Infrared LED (IR LED) transmitter for uplink communication. This system achieved a 500 kbps data rate over a 110 cm distance. Li-Fi technology was implemented in home automation by [14] in a cost-effective system for improved safety and comfort. Their design used a Raspberry Pi as the control unit and an Arduino as a transmitter for the communication with electronic devices. Highlighting the potential of Li-Fi technology for data speed, Karthik et al. [15] implemented a system for fast data transmission. Data from a PC is modulated by an Arduino UNO and transmitted via LED to a photodiode receiver connected to another Arduino UNO. Furthermore, Shanthi et al. [16] investigated underwater Li-Fi communication using Arduino UNO for data modulation and reception, with a solar panel on the receiver side. These various studies highlighted the diverse applications and continuing development of Li-Fi technology. Li-Fi technology boasts high security, but researchers have implemented additional encryption methods to enhance data protection. Chergui and Abdesselam [17] implemented a Li-Fi system using Arduino boards and used the widely recognised and secure Advanced Encryption Standard (AES) algorithm to encrypt text data before the transmission. In the same way, Kaftannikov et al. [18] designed a system to exchange data using Li-Fi technology for mobile phones, which included a custom algorithm for data formatting and XOR-based encryption, and decryption. Furthermore, Aldolimi et al. [19] used AES for data encryption in their Li-Fi system, using the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol for secure key generation. These studies highlighted the continuing efforts to strengthen Li-Fi security by implementing different encryption algorithms. Researchers have also increased data protection for the Internet of Things (IoT) by improving traditional encryption methods such as Tiny Encryption Algorithm (TEA) [20]. They proposed a novel tiny symmetric encryption algorithm (NTSA) that makes text file transfers over IoT networks more secure by dynamically adding new key confusions for every encryption round. Sylla et al. [21] proposed a secure context management system for context-aware security and privacy in smart cities. They implemented the Device Trust Management (DTM) module of the Context-Aware Security and Privacy as a Service (CASPaaS) architecture. It secured context exchange using lightweight hybrid encryption and managed trust with Bayesian networks and fuzzy logic techniques. Researchers have studied the Li-Fi technology in industrial automation. Mukku et al. [22] implemented a system, that used Li-Fi technology, in an Industry 4.0 learning laboratory. They used Radio Frequency Identification (RFID) reader and Li-Fi transceivers for product tracking. Georlette et al. [23] proposed a system that used Li-Fi technology for downlink communication to control Automated Guided Vehicles (AGVs) in a fruit factory, and infrared technology was used for uplink communication. Kurian et al. [24] focused on safety applications, designing a system based on Li-Fi technology that used various sensors to detect hazards and transmits data for worker notification. These studies highlighted the diverse applications of Li-Fi in industrial automation, from product tracking and communication to real-time safety monitoring.

Building upon the literature survey presented in Table 2, a research gap exists in that no previous research addresses real-time security threats in Li-Fi technology when the factory devices communicate with each other using Li-Fi technology. Security threats are attacks on the data sent by the Li-Fi technology that leak through the holes in the building, thus controlling factory equipment and causing a malfunction inside the factory that may deal with toxic or explosive materials. So, this article aims to address this vulnerability and contribute to secure the data transmission. A novel multi-mode approach is proposed that tackles security risks associated with wall leakage in the industrial field when it uses Li-Fi technology to communicate. This approach provides flexibility through three data transmission modes, which are an unencrypted mode, an encrypted mode with the RC4 algorithm, and an encrypted mode with the ARC4 algorithm.

3. Proposed algorithm and system

The proposed system has a central controller, transmitter, receiver, and objects as shown in Figure 2. The central controller acts as the main unit, where it manages the communication between the transmitter and the receiver. It determines what the data importance. The transmitter transmits data via Free Space Optical (FSO), where the transmitter uses visible light to send data. The receiver picks up the transmitted data from

Ref.	Microcontroller	Data Rate	Use	Distance	Encryption
[13]	ZYBO Zynq-7000	500 kbps	Indoor communication	110 cm	No
[14]	Arduino board	38 kbps	Indoor communication	10 m	No
[15]	Arduino UNO	90 Mbps	Data transmission	–	No
[16]	Arduino UNO	1 Gbps	Underwater communication	6 m	No
[17]	Arduino UNO	250 kbps	Text transmission	2 m	Yes
[18]	STM microcontroller	–	Data transmission	2 m	Yes
[19]	–	–	Text transmission	–	Yes
[20]	–	–	Internet of things	–	Yes
[21]	–	–	Smart city	–	Yes
[22]	Arduino Mega 2560	1 Gbps	Industrial automation	68 cm	No
[23]	Raspberry Pi 4	400 kbps	Fruit industry	–	No
[24]	Arduino UNO	1.5 Mbps	Industrial safety	10 m	No

TABLE 2. The survey summary of the Li-Fi technology and when they get transmitted.

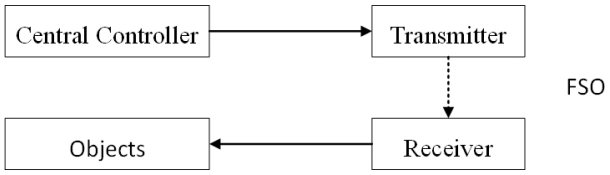


FIGURE 2. General architecture.

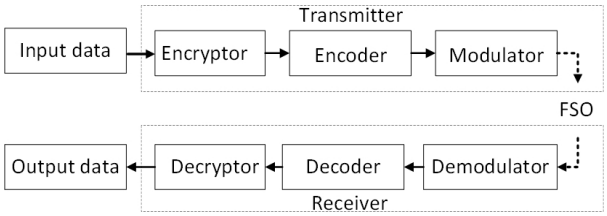


FIGURE 3. Data Transmission Structure with FSO.

the FSO. The object represents the data destination, interacting with the transmitter and receiver under the supervision of the central controller where the transmitter sends data to the receiver via the FSO. In this work, a PC acts as the central controller, sending data to a robotic arm via a transmitter built from Arduino Mega and a laser module, while data is received by another Arduino Mega with a laser sensor, as will be explained in the implementation section. The transmitter has an encryptor, an encoder, and a modulator. The encryptor encrypts data according to the selected mode for encryption by the user. The encryption modes are unencrypted mode, RC4 mode, and ARC4 mode where, these modes address concerns in [25]. The unencrypted mode is for a safe environment, and RC4 mode is for a normal environment, and the ARC4 mode is for a dangerous environment. A dangerous environment is an environment where the transfer contains sensitive data, hacking of which could cause a disaster. A normal environment is an environment where the transfer contains important data, and a security breach could lead to data theft. A safe environment is an environment where the transfer contains unimportant data which would not cause harm when hacked. The encoder encodes encrypted data according to the American Standard Code for Information Interchange (ASCII) to convert it to an 8-bit binary form. The modulator converts encoded data into pulse signals to turn on and off the visible light. The receiver has a demodulator, a decoder, and a decryptor. The demodulator converts pulse signals into encoded data after sensing the visible light in the FSO by the receiver. The decoder decodes the encoded data according to ASCII into encrypted data. The decryptor decrypts the encrypted data according to the selected encryption mode by the user. Figure 3 schematics the parts of the transmitter and receiver in this work.

4. IMPLEMENTATION

In this section, the materials used to implement the prototype are explained. The encryption modes used to secure the transmitted data are then explained. The section concludes with an illustration of how the components of the prototype are linked.

4.1. MATERIALS

The proposed system was implemented as a physical prototype. The proposed system consists of two parts, which are the transmitter side and receiver side. The transmitter side has a laser module and an Arduino Mega 2560 connected to the central controller, which is a Personal Computer (PC). The receiver side has a laser sensor and an Arduino Mega 2560 connected to the object, which is a robotic arm, as shown in Figure 4.

The PC transfers a movement command into Arduino Mega 2560, which acts with the laser module as a Li-Fi sender. The incoming movement command is then encrypted, encoded, and modulated using the Li-Fi sender to a pulse signal, which is transferred to the laser module. The laser module turns ON and OFF according to the pulse signal and the laser light spreads in the Free Space Optical (FSO). The laser sensor senses the laser light at the FSO to form a pulse signal and transfer it into Arduino Mega 2560 on the

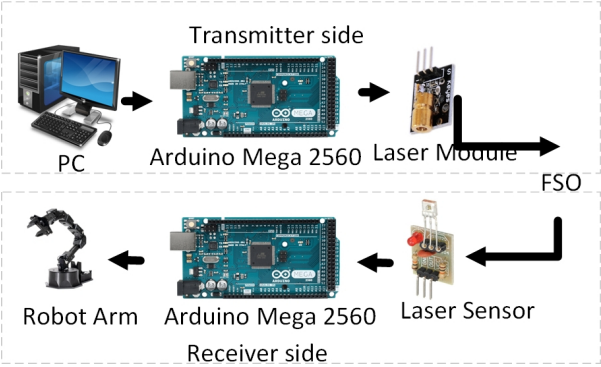


Figure 4. The proposed system.

Command	Explain
r030	Turn to the right by 30°.
r100	Turn to the right by 100°.
r180	Turn to the right by 180°.
l030	Turn to the left by 30°.
l100	Turn to the left by 100°.
l180	Turn to the left by 180°.
u030	Put the hand up hand by 30°.
u100	Put the hand up hand by 100°.
u180	Put the hand up hand by 180°.
d030	Put the hand down by 30°.
d100	Put the hand down by 100°.
d180	Put the hand down by 180°.

Table 3. Examples of commands to move robotic arm.

receiver side. The Li-Fi receiver records the binary of the pulse signal, demodulates, decodes, and decrypts it to get the movement command. The movement command is then transferred into the robotic arm. The robotic arm moves in a specific direction and degree according to the incoming movement command. The robotic arm implements its task depending on the incoming movement commands from the PC. The movement commands are instructions designed to fit the robotic arm movement capabilities. The movement commands consist of a letter and a direction degree, where the letter indicates the direction of movement, and the direction degree is the required displacement. The movement command is four digits which are one for a letter of the direction and three for the degree as shown in the examples in Table 3. The letters used in our system are r, l, u, d, o, and c which mean right, left, up, down, open, and close. l100 and r030 are examples of movement commands. l100 means to turn the robotic arm to the left by 100° while r030 means to turn the robotic arm to the right by 30°.

4.1.1. Arduino Mega 2560

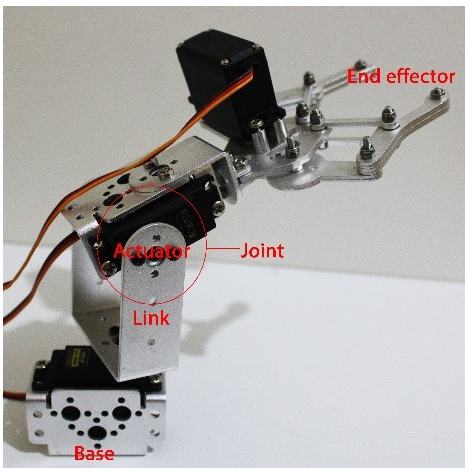
The Arduino Mega 2560 is a versatile programmable microcontroller board with the ATmega2560 microcontroller [26]. It boasts robust features, including 54 digital input/output pins, 16 analog inputs, and 4 Universal Asynchronous Receiver-Transmitter (UART), Inter-Integrated Circuits (I2C), and Serial Peripheral Interface (SPI) ports to connect and interact with various sensors and devices. 15 of the 54 digital input/output pins support Pulse Width Modulation (PWM). These pins are used to control the servo motors of the robotic arm. Arduino Mega 2560 offers 8 KB of Static Random-Access Memory (SRAM), 256 KB of Flash memory for program me storage, and 4 KB of Electrically Erasable Programmable Read-Only Memory (EEPROM) for non-volatile data storage [27].

4.1.2. Robotic arm

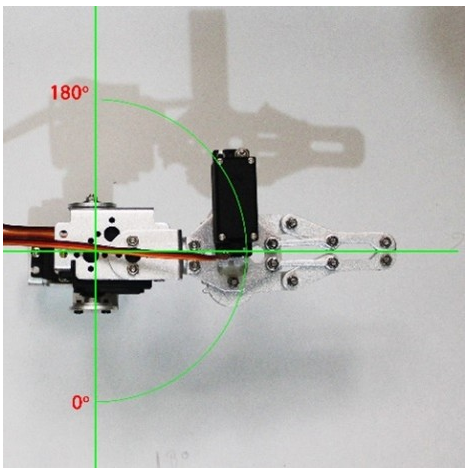
A robotic arm is a programmable machine that mimics a human arm. It has several parts: a base for stability, joints for movement, and a gripper or tool at the end as shown in Figure 5a. These arms are used in many industries for tasks that must be done quickly and precisely [28]. The proposed system uses a 2-DOF robotic arm shown in Figure 5a. This robotic arm comprises three servo motors: two for yaw and roll movements, and one to open and close the hand. Yaw motion allows the arm to rotate horizontally between 0° and 180° as shown in Figure 5b. Similarly, roll enables the arm to rotate vertically between 0° and 180°, with clockwise movement lowering the hand and counterclockwise raising it as shown in Figure 5c. The final servo controls the hand aperture by opening or closing it, through a 0° to 180° rotation range as shown in Figure 5d. However, to prevent motor damage, the operational range for hand control must be limited to a range between 0° and 60°.

4.1.3. Transmitter

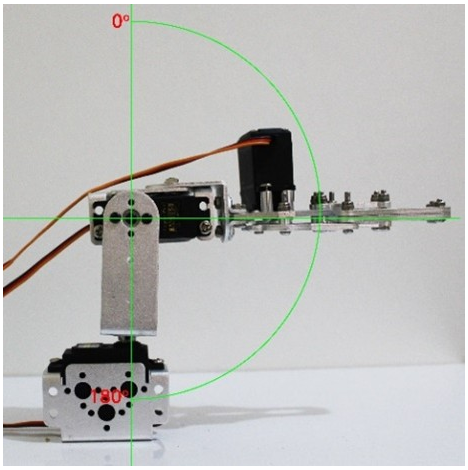
The transmitter comprises an Arduino Mega 2560, laser module, and LCD as shown in Figure 6a. The transmitter connects with the PC via UART where the user types the movement command on the PC. These bytes of movement command are put in stacks to reverse bitwise to form a transmission packet. The packet structure varies depending on the mode selected by the user. In the initial connection, the packet comprises six bytes in the unencrypted mode. These bytes are: one for connection establishment, one for encryption mode indication, and four for the movement command. In the unencrypted mode, there is no secret key so there are no bytes to indicate the secret key in the transmission packet. The transmission frame without bytes for the secret key is shown in italics in Figure 7a. Conversely, the encrypted mode packets have nine bytes. These bytes are: one for connection establishment, one for encryption mode, three bytes for a randomly generated secret key, and four for the movement command as shown in Figure 7a. The transmitter encrypts the movement command using RC4 or ARC4, if the mode has an encryption algorithm. Finally, each digit of the frame is converted to binary and put into stacks. After establishing the



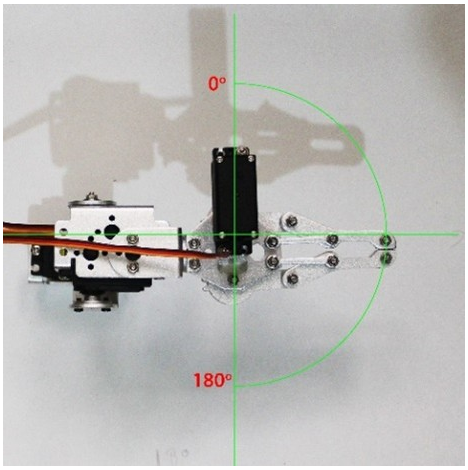
(A). 2-DOF Robot Arm.



(B). The yaw movement.

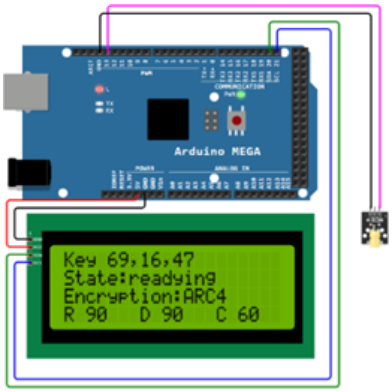


(C). The roll movement.

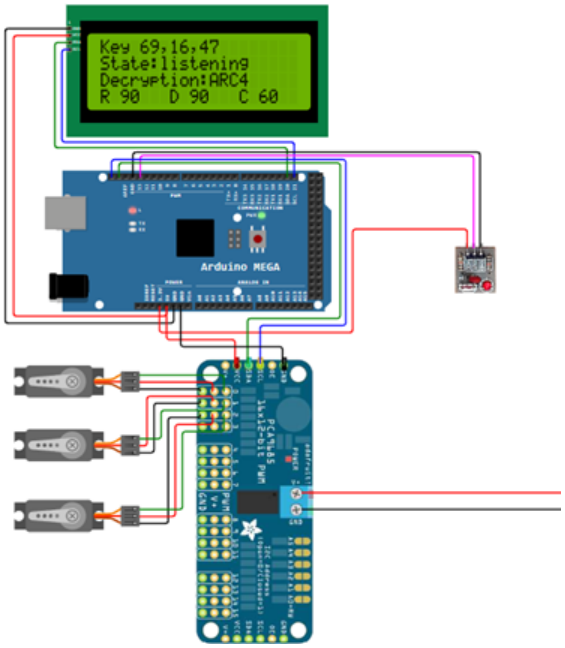


(D). The opening and closing hand.

FIGURE 5. The robotic arm used in this work.



(A). The transmitter.



(B). The robotic arm linked to the receiver.

FIGURE 6. The components of our proposed model.

1	One byte	Connection Establishment
2	One byte	Encryption Mode
:	*Three bytes*	*Secret Key*
9	Four bytes	Movement Command

(A). Frame formats for establishing the connection.

1		
:	Four bytes	Movement Command
4		

(B). Frame format of the established connection.

FIGURE 7. Frame formats of data transmission.

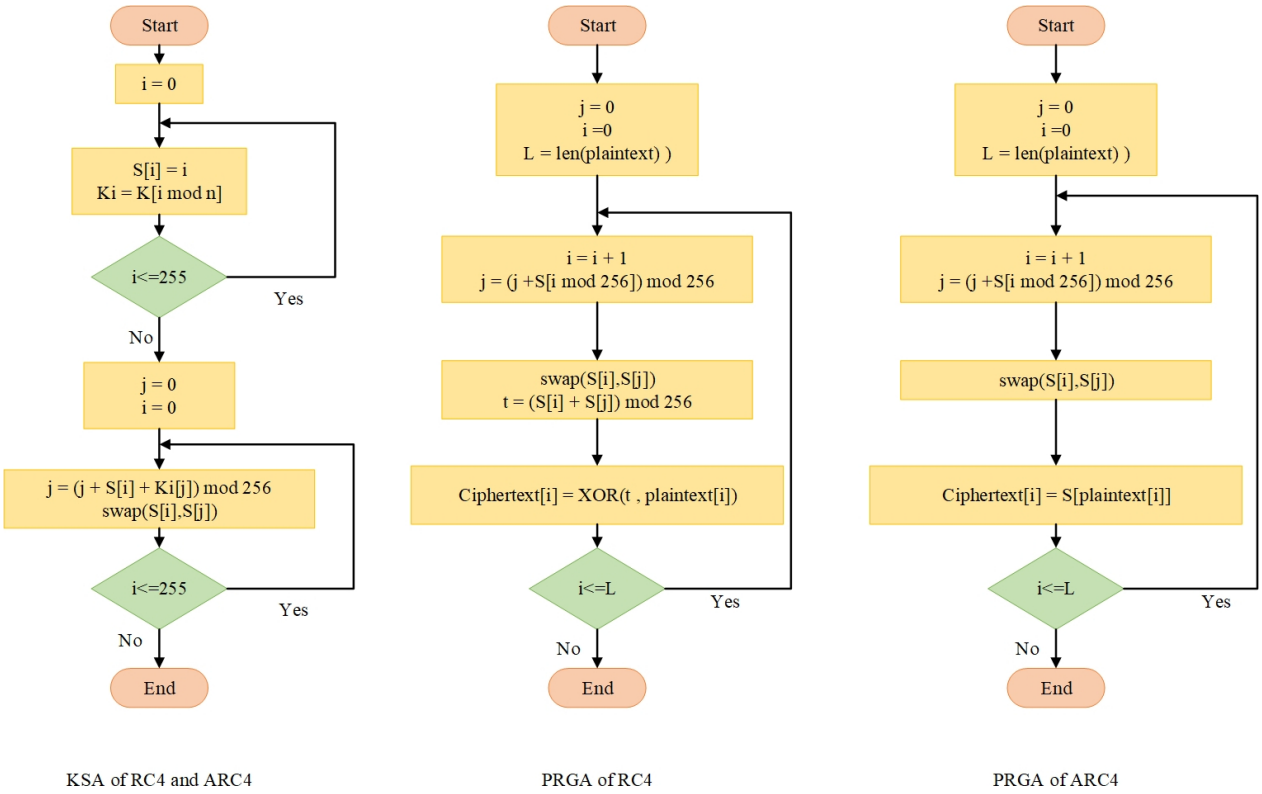


FIGURE 8. Flowcharts of RC4 and ARC4 algorithms.

connection, the transmission packet has four bytes, which are for the movement commands as shown in Figure 7b. The stacks of data are transmitted to FSO using the ON-OFF of the laser module.

4.1.4. RECEIVER

The receiver comprises an Arduino Mega 2560, laser sensor, and LCD as shown in Figure 6b. The receiver uses a buffer of eight bits in size to save the received data stream. Incoming data bits are stored in the buffer until a full byte is received. The initial connection handshake involves receiving and decoding the bytes for establishing the connection and encryption mode. If the encryption using RC4 or ARC4 is selected, three additional bytes containing the secret key are received. Finally, the four bytes of the movement command are received. Subsequent receiving during an established connection only includes the four bytes of the movement command. The receiver decodes the received data and decrypts it, if the mode has an encryption algorithm to form the movement command. The receiver moves the robotic arm according to the movement command, making the robotic arm move at a certain angle and direction.

4.2. PROPOSED ENCRYPTION METHODS

There are three modes in our proposal to transfer movement commands. These modes are movement commands without encryption, movement commands with encryption by the Rivest Cipher 4 (RC4) algorithm, and movement commands with encryption by the Advanced Rivest Cipher 4 (ARC4) algorithm. In the case of the RC4 algorithm, the key is generated in the key-scheduling algorithm (KSA), and data are encrypted in the pseudo-random generation algorithm (PRGA) as shown in Figure 8. The ARC4, which is an improved version of RC4, was developed by [10]. The plaintext is the index of the S-array and the ciphertext is the content of the S-array containing PRGA, only developed in ARC4, as shown in Figure 8. This proposal allows sending data at different levels of security depending on the environment and the importance of data. When the environment is safe and the data being transmitted are unimportant, the unencrypted mode is used. When the environment is normal and the data being transmitted are important, the RC4 encryption mode is used. When the environment is dangerous and the data being transmitted are sensi-

Character	Byte	Symbol	Explain
0	110000	None	Unencrypted Mode.
1	110001	RC4	Encrypted Mode by RC4.
2	110010	ARC4	Encrypted Mode by ARC4.

TABLE 4. Modes of our proposal.

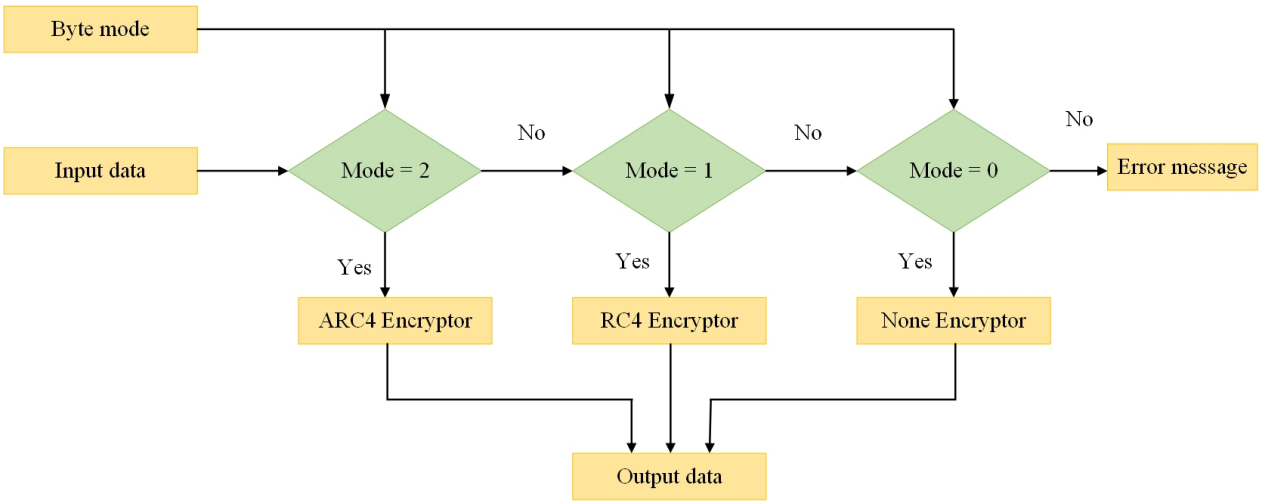


FIGURE 9. Flowchart of using the modes in our system.

tive, the ARC4 encryption mode is used. Determining the importance of the data and the type of environmental risk depends mainly on the user. Information on the mode used by a user is provided to the receiver by sending a byte at the beginning of the communication. Table 4 illustrates the byte transmitted by the transmitter to determine the encryption mode. If the mode uses encryption, the transmitter randomly generates a secret key and sends it after the byte of the encryption mode. The encryption process takes place before sending, and the decryption process takes place after receiving. These modes are in the transmitter and the receiver and process input data to output data as shown in Figure 9. On the transmitter side, the input data is the original data, and the output data can be encrypted using the RC4, or ARC4, or unencrypted. On the receiver side, the input data can be encrypted using RC4, or ARC4, or unencrypted, while the output data is unencrypted or decrypted.

5. RESULT

In this section, the results of the proposal implementation are presented. First, the implemented prototype is presented. Second, the transmitted and received data from the transmitter and the receiver are presented. Finally, experiments are conducted to confirm the robustness of our proposal.

5.1. PROPOSED ENCRYPTION METHODS

Figure 10 shows the final prototype for the transmitter and the receiver connected to the robotic arm. A user types the movement commands on a PC that transfers it to an Arduino Mega 2560 programmed as

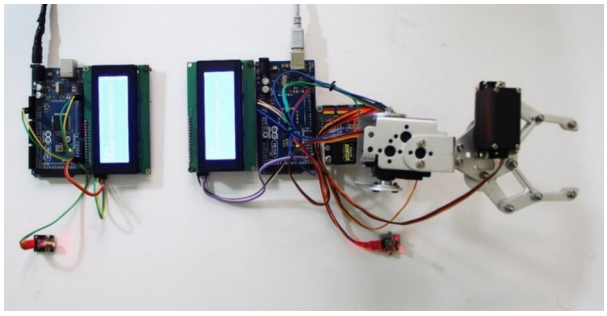


FIGURE 10. The implemented prototype.

the transmitter. The transmitter encrypts the movement commands before transmitting them if the mode has an encrypted algorithm. The transmitter then transmits it as binary data via a laser module. The laser modulates its light intensity based on the binary sequence illuminating the FSO. On the receiver side, another Arduino Mega 2560 programmed as a receiver detects the visible light via a laser sensor and decodes the received binary data. Decryption takes place if the mode uses encryption. Finally, the receiver transfers the decoded movement command to the robotic arm, instructing it to move.

5.2. DESIGNED GRAPHICAL INTERFACE USER

The user interacts with a graphical interface designed in C# language to control the movement of the robotic arm. The implemented graphical interface user is shown in Figure 11. After linking the Li-Fi sender to the PC, the user must choose the port through which the Li-Fi sender is connected, which will appear in the

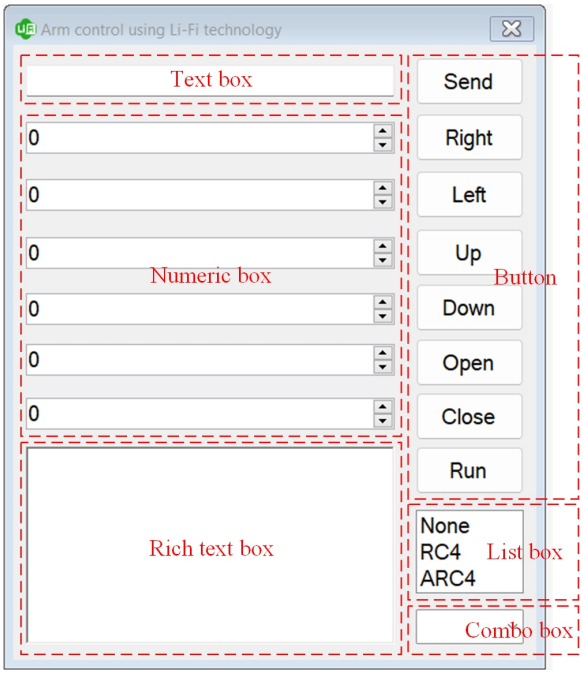


Figure 11. Designed a graphical interface to control the robotic arm.

combo box of the designed graphical interface, and the user must choose the mode of encryption from the list in the designed graphical interface. The user can then use the designed graphical interface to control the movement of the robotic arm. There are three cases for using the designed graphical interface. In the first one, the user enters the movement command in the text box and then clicks the "send" button. For example, the user enters r100 in the text box and then clicks the "send" button. The robotic arm will turn to the right by 100°. In the second one, the user enters a degree in the numeric box, which is located next to the button "direction required", and then clicks the button "direction required". Also, the user can use the buttons in the numeric box to increase or decrease the degree. For example, the user enters 100 in the numeric box, which is located next to the "Right" button, and then clicks the "Right" button. The robotic arm will turn to the right at 100°. In the third one, the user enters the movement commands in the rich text box and then clicks the "Run" button to send a series of movements. For example, the user enters r100, l020, and u090 in the rich text box and then clicks the "Run" button. The robotic arm will turn to the right by 100°, then to the left by 20°, and then upwards by 90°.

5.3. Proposed system states

The transmitter transmits a movement command to the receiver that decodes the movement command into rotation for the servo motor. There are six states of the transmitter of our system, which are the disconnection state, connecting state, sending state, sent state, and readying state. There are six states of the receiver of our system, which are the disconnection state, connecting state, receiving state, moving state, and listening state. In the disconnection state, there is no connection between the transmitter and the receiver, while in the connecting state is trying to connect between the transmitter and the receiver. In the sending state, data are being transmitted, while in the sent state, the data have been sent. The transmitter is ready to send in the ready state. In the receiving state, the data are being received by the receiver. In the moving state, the receiver decodes the movement command and rotates the servo motor. In the listening state, the receiver is ready for data transmission. Figure 12 shows these states for a movement command r100 sent in RC4 mode.

5.4. Transmitted and received data

This section displays the data and signals transmitted using the laser module and received using the laser sensor for the three modes. The Hantek6004BC oscilloscope was used to collect the transmitted and received data and signals. Microsoft Excel was then used to draw signals. Table 5 presents examples of the transmitted and received data for the three modes. The laser module turns ON when it transmits the binary digit 1 and turns OFF when it transmits the binary digit 0. The laser sensor records the binary digit 0 when it detects laser light and records the binary digit 1 when it does not detect any laser light. Thereby, the received data is the inverse of the transmitted data. Therefore, each bit is inversed before the movement command is decoded.

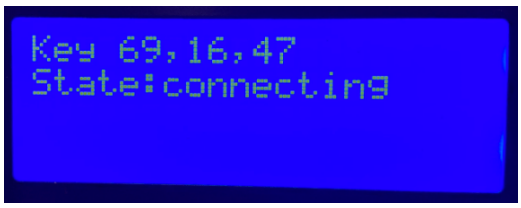
The signal is transmitted using the laser module by turning it ON and OFF, and the signal is received using the laser sensor by sensing a laser light in the FSO. The laser module turns ON at 3.89 volts when it transmits the digit of binary 1 and OFF at 0 volts when it transmits the digit of binary 0. The laser sensor outputs 3.77 volts when detecting the laser light and outputs 0 volts when not detecting laser light. The transmitting and receiving signals for the unencrypted mode are shown in Figure 13. The calibration process was used before sending the signal and after receiving the signal. The transmitter was calibrated by a process where the binary digit 1 is represented by 3.89 volts and binary digit 0 as 0 volts. The receiver was calibrated by a process that represented 3.77 volts as the binary digit 1 and 0 volts as the binary digit 0. The calibrated transmitting and receiving signals for the three modes are shown in Figure 14.

5.5. Evaluation

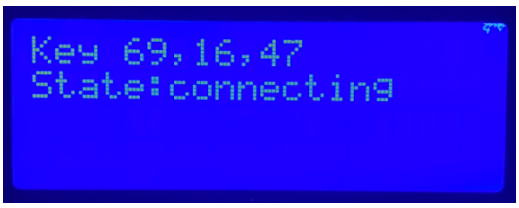
In this section, experiments are conducted to confirm the robustness of our proposed method. Firstly, the response of the robotic arm to the movement commands is tested. Secondly, the encryption in our system is tested by connecting a servo motor to the receiver and adding a hacker who connects to the servo motor.

(A). Unconnected sender.

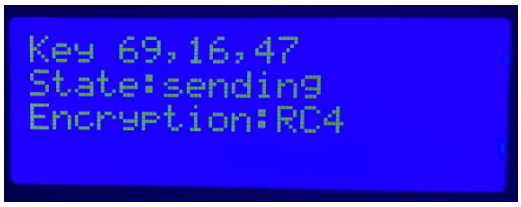
(B). Unconnected receiver.



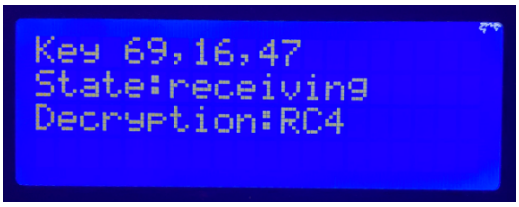
(C). Sender connecting.



(D). Receiver connecting.



(E). Data sending.



(F). Data receiving.

(G). Data sent.

(H). Servo motor rotating.

(I). Sender ready to send.

(J). Receiver ready to receive.

FIGURE 12. States of prototype.

Command		Transmitted data	Received data
r100	None	01001110100011000000110000001100	10110001011100111111001111110011
	RC4	10001110110000000110010000001000	01110001001111111001101111110111
	ARC4	11100100111000010010000100100001	00011011000111101101111011011110
r180	None	01110010001100010011100000110000	10001101110011101100011111001111
	RC4	11111000101110010001000011011000	00000111010001101110111100100111
	ARC4	00100111100001111000100010000100	11011000011110000111011101111011
l135	None	01101100001100010011001100110101	10010011110011101100110011001010
	RC4	01101111000000110010010100010101	10010000111111001101101011101010
	ARC4	10111010100001110101010000101001	01000101011110001010101111010110
l090	None	01101100001100000011100100110000	10010011110011111100011011001111
	RC4	11111111100111000111100101111010	00000000011000111000011010000101
	ARC4	10111010100001000101100110000100	01000101011110111010011001111011

TABLE 5. The transmitted and received data.

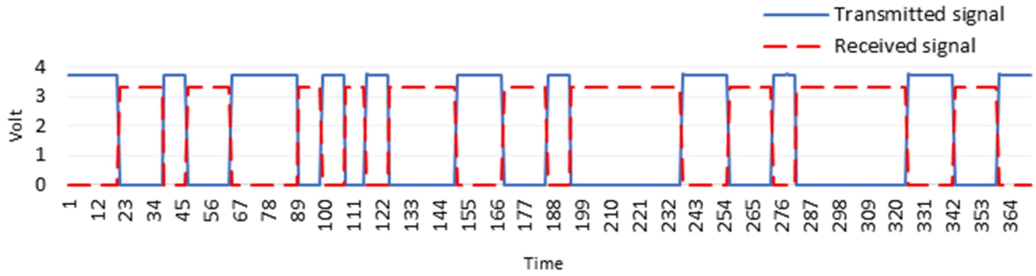


Figure 13. Pulse signals.

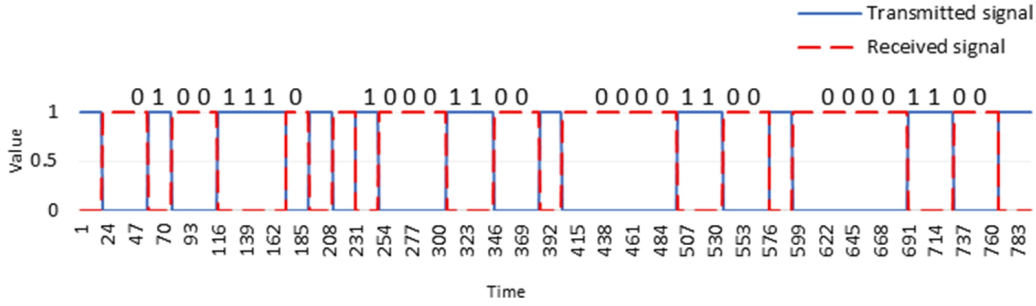
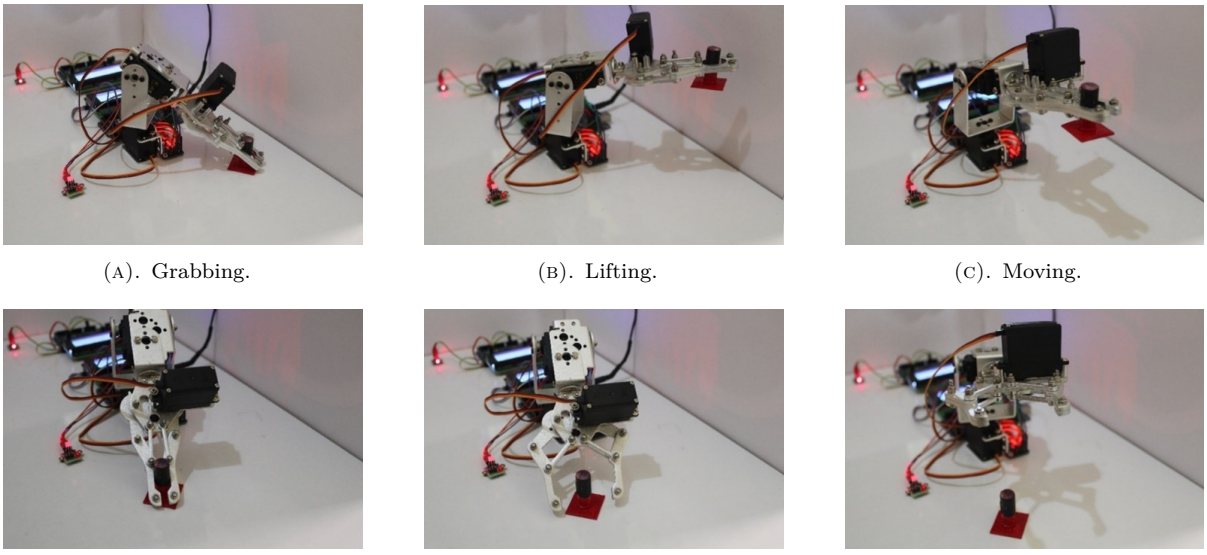


Figure 14. Calibrated pulse signals



(a). Grabbing. (b). Lifting. (c). Moving.

(d). Lowering. (e). Opening. (f). Abandoning.

Figure 15. The movement states of the robotic arm.

5.5.1. Robotic arm testing

The robustness of the response in the proposed system is presented in this section. The received side received movement commands and decoded and decrypted them to move the robotic arm that performed its task. The movement commands were used for the procedure to move a cylinder in the unencrypted mode. The movements were grabbing, lifting, moving, lowering, opening, and abandoning as shown in Figure 15.

5.5.2. Encryption testing

The robustness of the encryption of the proposed system was evaluated through a series of tests. In the unencrypted mode, the user and the hacker successfully decoded the movement command, resulting in identical movements as shown in Figure 16. Conversely, when RC4 encryption was implemented, only the user decrypted the command, causing the servo motor of the user to move as shown in Figure 17. The inability of the hacker to decode the encrypted signal rendered its servo motor motionless as shown in Figure 18. Similar results were obtained for the ARC4 encryption as shown in Figure 17, demonstrating the effectiveness of both encryption algorithms in protecting data transmission from unauthorised access.

5.5.3. Performance evaluation

The Peak Signal-to-Noise Ratio (PSNR), which is an expression for measuring the encrypted data relative to

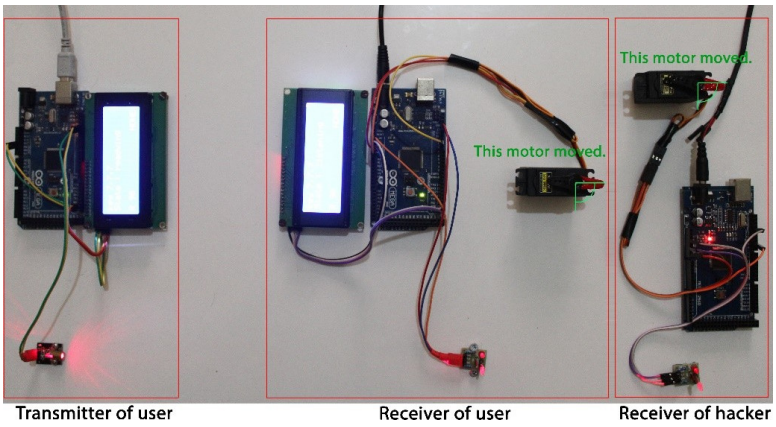


FIGURE 16. The prototype testing in unencrypted mode.

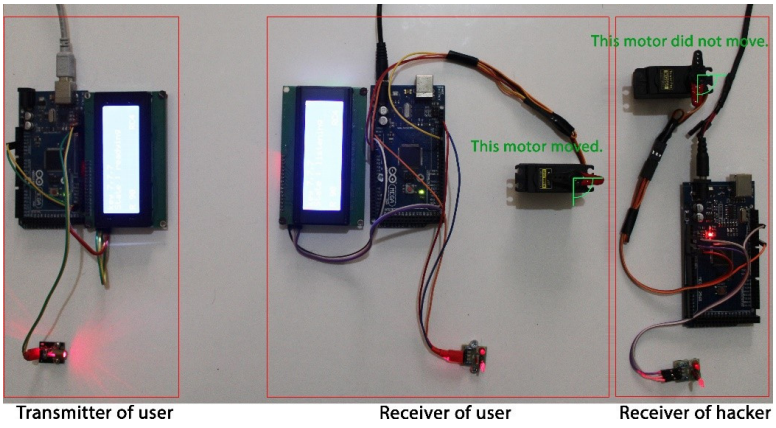


FIGURE 17. The prototype testing in RC4 mode.

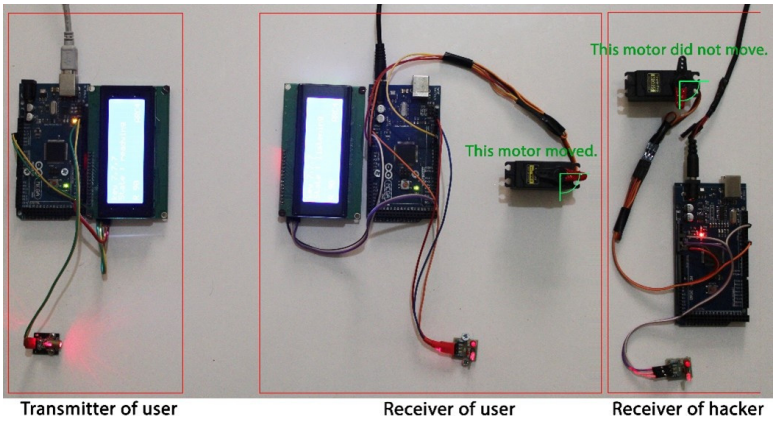


FIGURE 18. The prototype testing in ARC4 mode.

the original data, is one for evaluating the performance of our work. The lower the value of PSNR, the random the data in the channel. Due to this randomness, the hacker fails to interpret the command data in the channel. The Mean Squared Error (MSE) is the mean of the squared changes between the encrypted data and the original data. Equation (1) and Equation (2) are the mathematical expressions for MSE and PSNR, respectively [25]:

$$\text{MSE} = \frac{1}{L}\sum_{r=0}^{L-1}(E(r) - O(r))^2, \tag{1}$$

$$\text{PSNR}(db) = 10\log_{10}\frac{(2^k-1)^2}{\text{MSE}}, \tag{2}$$

where the values of encrypted and original data are $E(r)$ and $O(r)$, respectively. The length of data is denoted by the letter L. The number of bits in a byte is k, which is equal to 8. The value of PSNR for ARC4 is smaller than the value of PSNR for RC4 mode because the transmitted data in ARC4 mode is more random, and hence, the security of the encrypted data is higher. Furthermore, the small value of PSNR for the ARC4 mode means that ARC4 is more secure because there is a high random distribution when the

Command	ARC4 mode		RC4 mode	
	MES	PSNR	MES	PSNR
r100	16 580.75	5.9347	2227.75	14.652
r180	17 574.75	5.6819	1787.75	15.607
l135	10 343.25	7.9842	1753.75	15.691
l090	11 449	7.5431	1683.75	15.868

Table 6. Compare results.

data is transmitted. Table 6 shows the values of MSE and PSNR for the RC4 mode and the ARC4 mode.

6. Conclusion

This article presents a novel multi-mode approach to secure Li-Fi communication with protection against wall-leakage vulnerabilities in an industrial environment. The approach offers flexibility to transmit data in three modes, depending on the security needs of the environment are the importance of the transmitted data. The three modes are an unencrypted mode, which is used in safe environments and transmits unimportant data, a mode encrypted by RC4, which is used in normal environments and transmits important data, and a mode by ARC4, which is used in dangerous environments and transmits sensitive data. The proposed system could be expanded for future development to incorporate additional encryption algorithms. Additionally, implementing the Diffie-Hellman key exchange would improve the security during secret key transmissions. Lastly, incorporating multiple laser modules at various points would mitigate data transmission disruption caused by barriers obstructing the single-point module.

References

[1] O. Faruq, K. R. S. Rahman, N. Jahan, et al. Li-Fi technology-based long-range FSO data transmit system evaluation. *Sustainable Engineering and Innovation* **5**(1):85–98, 2023. https://doi.org/10.37868/sei.v5i1.id192

[2] M. Mokayef, M. G. Reddy, M. H. D. A. Summakieh, A. Mohammadpour. Li-Fi technology for enhanced communication and safety in coal mining. *Journal of Robotics, Networking and Artificial Life* **10**(2):170–178, 2023. https://doi.org/10.57417/jrnal.10.2_170

[3] A. Agarwal, C. Mohanta, G. Misra. Li-Fi technology: Principle, future scope, challenges and applications. *American Journal of Electrical and Electronic Engineering* **10**(1):1–5, 2022. https://doi.org/10.12691/ajeee-10-1-1

[4] H. K. Yu, J. G. Kim. Smart navigation with AI engine for Li-Fi based medical indoor environment. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 195–199. 2019. https://doi.org/10.1109/ICAIIC.2019.8669041

[5] S. M. Umran, S. Lu, Z. A. Abduljabbar, V. O. Nyangaresi. Multi-chain blockchain based secure data-sharing framework for industrial IoTs smart devices in petroleum industry. *Internet of Things* **24**:100969, 2023. https://doi.org/10.1016/j.iot.2023.100969

[6] E. Ramadhani. A mini review of LiFi technology: Security issue. *International Journal of Computer and Information System (IJCIS)* **3**(3):90–93, 2022. https://doi.org/10.29040/ijcis.v3i3.74

[7] Sulistiyanto, I. Satriadi, A. Rahman. Electronic archive design with RC4 cryptographic based file security. *Journal of Computer Networks, Architecture and High Performance Computing* **6**(1):34–44, 2024. https://doi.org/10.47709/cnahpc.v6i1.3298

[8] A. F. Ghaleb, A. A. Oglah, A. J. Humaidi, et al. Optimum of fractional order fuzzy logic controller with several evolutionary optimization algorithms for inverted pendulum. *International Review of Applied Sciences and Engineering* **14**(1):1–12, 2023. https://doi.org/10.1556/1848.2021.00375

[9] M. A. Hasan, A. A. Oglah, M. J. Marie. Packet loss compensation over wireless networked using an optimized FOPI-FOPD controller for nonlinear system. *Bulletin of Electrical Engineering and Informatics* **11**(6):3176–3187, 2022. https://doi.org/10.11591/eei.v11i6.4345

[10] M. M. Msallam, R. Samet. An advanced Rivest Cipher 4 algorithm to transfer fast and secure data using Li-Fi technology. In *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)*, pp. 194–199. IEEE, 2023. https://doi.org/10.1109/ICSET59111.2023.10295143

[11] S. Chugh, Kamal. Securing data transmission over wireless LAN (802.11) by redesigning RC4 algorithm. In *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 1436–1441. IEEE, 2015. https://doi.org/10.1109/ICGCIoT.2015.7380693

[12] R. Alsharida, M. Hammood, M. A. Ahmed, et al. RC4D: A new development of RC4 encryption algorithm. In *Selected Papers from the 12th International Networking Conference*, pp. 19–30. Springer, 2021. https://doi.org/10.1007/978-3-030-64758-2_2

[13] S. Fuada, T. Adiono, F. Ismail, E. Setiawan. Prototyping the Li-Fi system based on IEEE 802.15.7 PHY.II.1 standard compliance. *Journal of Communications* **15**(6):519–527, 2020. https://doi.org/10.12720/jcm.5.6.519-527

[14] P. Šuľaj, R. Haluška, L. Ovseník, et al. An example of Li-Fi technology implementation for home automation. In *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, pp. 183–187. IEEE, 2018. https://doi.org/10.1109/DISA.2018.8490607

[15] V. Karthik, K. Balashanmugam, S. Abithsingh, et al. High speed transmission of data or video over visible light using Li-Fi. In *2022 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, pp. 1–6. IEEE, 2022. https://doi.org/10.1109/ICACTA54488.2022.9753036

[16] T. Shanthi, P. Shalini, S. Shahul Hameed. Communication between submarines using Li-Fi technology. In *2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*, pp. 1–4. IEEE, 2023. https://doi.org/10.1109/ICRASET59632.2023.10420343

[17] S. Chergui, S. Abdesselam. Design and realization of a visible light communication system for Li-Fi application. In *2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)*, pp. 30–35. IEEE, 2020. https://doi.org/10.1109/CCSSP49278.2020.9151780

[18] I. L. Kaftannikov, A. V. Kozlova, A. D. Khlyzov. Prototype of a Li-Fi communication system for data exchange between mobile devices. In *2020 Global Smart Industry Conference (GloSIC)*, pp. 192–198. IEEE, 2020. https://doi.org/10.1109/GloSIC50886.2020.9267814

[19] W. S. Aldolimi, A. A. Hnaif, M. A. Alia. Light fidelity to transfer secure data using advanced encryption standard algorithm. In *2021 International Conference on Information Technology (ICIT)*, pp. 963–967. IEEE, 2021. https://doi.org/10.1109/ICIT52682.2021.9491769

[20] S. Rajesh, V. Paul, V. G. Menon, M. R. Khosravi. A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices. *Symmetry* **11**(2):293, 2019. https://doi.org/10.3390/sym11020293

[21] T. Sylla, M. A. Chalouf, F. Krief, K. Samaké. SETUCOM: Secure and trustworthy context management for context-aware security and privacy in the internet of things. *Security and Communication Networks* **2021**(1):6632747, 2021. https://doi.org/10.1155/2021/6632747

[22] V. D. Mukku, S. Lang, T. Reggelin, P. Reichardt. Design of a Li-Fi transceiver for distributed factory planning applications. In *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, pp. 188–197. Springer, 2021. https://doi.org/10.1007/978-3-030-85874-2_20

[23] V. Georlette, J. S. Melgarejo, S. Bette, V. Moeyaert. Automated guided vehicle controlled by Li-Fi: A study case. In *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pp. 53–58. IEEE, 2022. https://doi.org/10.23919/ICCAS55662.2022.10003807

[24] N. S. Kurian, R. Preetha, N. S. J. Joan, et al. Li-Fi based industrial safety module. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 17–22. IEEE, 2021. https://doi.org/10.1109/ICCMC51019.2021.9418429

[25] M. M. Msallam, R. Samet. A review of security methods in light fidelity technology. *Proceedings of Engineering and Technology Innovation* **27**:1–17, 2024. https://doi.org/10.46604/peti.2024.13149

[26] R. Anbalagan, M. Z. Hussain, D. Jayabalakrishnan, et al. Vehicle to vehicle data transfer and communication using Li-Fi technology. *Materials Today: Proceedings* **45**:5925–5933, 2021. https://doi.org/10.1016/j.matpr.2020.08.786

[27] H. Kareem, D. Dunaev. The working principles of ESP32 and analytical comparison of using low-cost microcontroller modules in embedded systems design. In *2021 4th International Conference on Circuits, Systems and Simulation (ICCSS)*, pp. 130–135. IEEE, 2021. https://doi.org/10.1109/ICCSS51193.2021.9464217

[28] M. Shyam, M. Amalasweena, S. Suvitha, et al. Intellectual design of bomb identification and defusing robot based on logical gesturing mechanism. In *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pp. 1–8. IEEE, 2023. https://doi.org/10.1109/ACCAI58221.2023.10201034