International Journal of Emerging Engineering and Technology (IJEET)
ISSN (e): 2958-3764
Volume: 2, Number: 1, Pages: 35- 45, Year: 2023

Lightweight Message Authentication Protocol for Low-Resource IoT Devices

Sadaf Hussain[1], Rabia Afzaal[1], Syeda Aina Batool[2]

[1] Faculty of Computer Science, Lahore Garrison University, Lahore, Pakistan
[2] Computer Science Department, Institute of Management Sciences, Lahore, 54000, Pakistan

Corresponding author: Sadaf Hussain (Email: sadafhussain@lgu.edu.pk)

Received: 12/01/2023, Revised: 22/03/2023, Accepted: 15/06/2023

Abstract— **The Internet of Things (IoT) is a system of physical objects attached to software and other technologies that allow them to connect to and exchange information between devices and systems over the Internet. There is a framework in the Internet of Things (IoT) in which devices are usually outfitted with wireless sensor nodes to connect several physical devices over the medium to obtain data without human intervention. Message Authentication in low-resource devices such as sensors is inefficient with heavy protocols because it quickly drains the battery. In our proposed scheme, we used CRC. We compared it with other hashing protocols to introduce a lightweight message authentication protocol for low-resource devices in various fields, such as healthcare and daily life gadgets.**

***Index Terms*— CRC, Hashing, IoT devices, Message Authentication, WBAN, WSN**

I. Introduction

IoT (Internet of Things) is a framework where several devices are installed with wireless sensors to connect physical devices over the internet mediums to generate, exchange, and move information without human collaboration [1]. Primitively, IoT was dubbed the "Internet of Everything". It can be categorized into 3 categories: M2M, People to People, and People to machines, interconnected via the internet to each other. The idea of IoT, from our day-to-day small devices to large-scale industrial systems, has enabled the devices to see the real world around them. IoT can be categorized into 3 categories: M2M, People to People, and People to machines, interconnected via the internet to each other. Though IoT is much more than these [1]. The IoT term can cover many technologies. "Wireless sensor networks, Cloud computing, web services, mobile internet, communication protocols, embedded systems, etc." are all IoT-enabling technologies. However, WSN is the core of IoT [2].

The use of such IoT applications is spreading throughout the world. Gartner predicts that the number of machine-to-machine (M2M) connections will rise from "5.6 billion in 2014 to 27 billion in 2024" [3]. As wearable devices enter the market, the wireless body area network (WBAN) is becoming a popular domain for IoT-connected healthcare applications [4]. The Internet of Things (IoT) is a popular technological concept that connects handlers and gadgets via wired and wireless technologies such as "Wireless Sensor Networks (WSNs), ZigBee, NFC, RFID, GPRS, LTE, and Bluetooth."[5]. Low energy wireless communications, intelligent sensing, bar codes", anywhere, anytime globally [6].

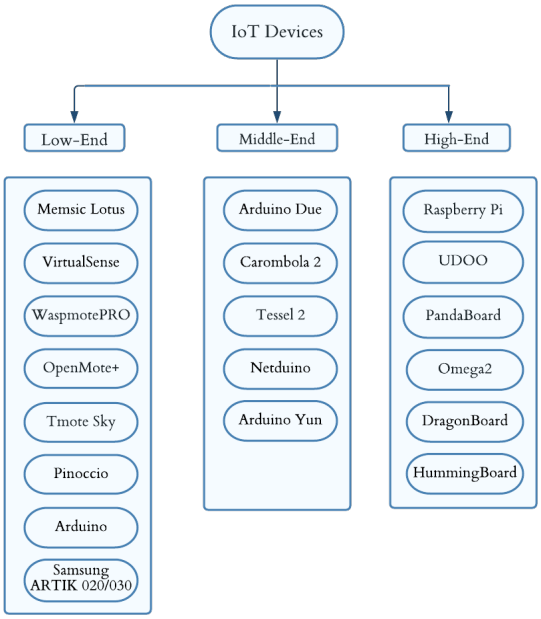

FIGURE 1. IoT Devices

Low-resource IoT devices refer to devices that are limited resources to run Traditional Operating systems such as

"Windows 10 or Linux OS" shown in Fig. 1.

For elementary sensing and stimulating applications, these devices are mass-produced. Very low-performance WSN OS or low-level firmware is used to program this low-resource device. Low-end devices are Wasp-mote PRO and Virtual Sense [7].

Low-resource IoT technologies such as WSN and wearable BAN face battery issues, i.e., limited battery power, sensor network nodes are deployed in an environment where we cannot change batteries often, such as in healthcare, replacing a battery means early death of the patient. Thus, these low-resource devices should be more energy-efficient as their battery life is critical for their users. Wearables, WSN, and WBAN are interchangeable terms.

Applications of IoT can be categorized into various domains. Such as "transportation, logistics, healthcare, smart environments, personal and social domain, and futuristics domain" [1]. The most important and immensely growing application of IoT is in healthcare.

A. Challenges to Low-Resource IoT Devices

In a diversified environment for IoT, there are many challenges to low-resource IoT applications stated [Fig 2]:

A. Power consumption
B. Security issues
C. Authentication
D. Identification,
E. Limited battery
F. Cost of performance
G. Memory space
H. Confidentiality
I. Access control
J. Privacy etc. [8]

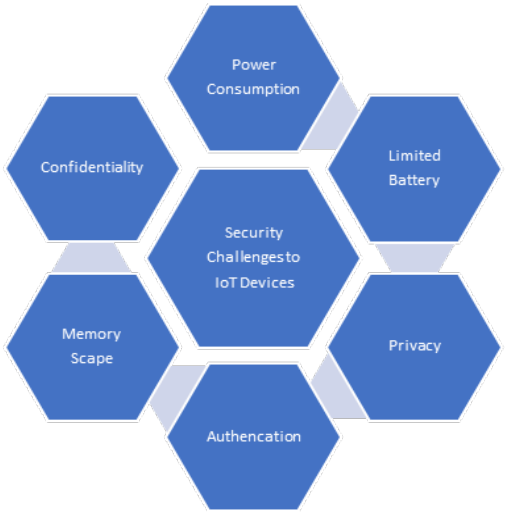

FIGURE 2. Challenges to IoT Devices

Issues and challenges like these seriously affect the performance of IoT, and a good system is always the least affected by challenges. Instead, these systems should take various countermeasures to avoid these threats in a real-time environment. Our main research focus is "message authentication" in low-resource IoT wireless devices.

B. Goals and Objectives of the Research

The study determines to progress lightweight energy-efficient message authentication protocols that consume energy as minimum as possible to ensure message integrity.

C. Contribution

Our contribution will be as follows:

Heavy message authentication protocols can drain the battery quickly, so we are introducing a lightweight message authentication protocol. It will increase a device's battery, hence increasing the life of the network in which it will be used. In critical systems like healthcare, the need to provide quality care to patients while dropping the costs of systems is a matter of discussion.

In our proposed scheme, we introduced a lightweight message authentication protocol based on CRC for low-end Devices.

II. Related Work

As the IoT grows, billions of "low-cost" devices will be interconnected, enabling users to browse network innovations. The value of network access obtained by these devices is also increasing, making each other a potential attack target. Low-cost Internet of Things devices typically have limited calculation, storage, and energy abilities. As a result, employing conventional cryptographic procedures to protect them is not always feasible.

In [9], this publication investigated whether a cryptographic algorithm CMAC and a KMAC meet resource-constrained IoT device limits. They compared the performance of built-in 65nm CMOS with 128-bit CMAC and 128-bit KMAC. As a result of their testing, the CMAC-128 is deemed more compact and stronger than the KMAC-128 at 1.2 volts and with a frequency greater than 1MHz.The authors [9] of this work standard cryptographically safe using the fewest resources possible. A new CRC-based message authentication method has been initiated, and a quantitative study of the security attained as a function of message and CRC sizes was conducted. The provided technique preserves most of the classic CRC's execution simplicity, except that the LFSR that does the decryption must have re-programmable connections. Like previously suggested cryptographically secure CRCs, the provided CRC identifies stochastic and intentional mistakes without increasing throughput. Its key benefit is that it can catch any 2-bit mistakes in a message, which is significant for technologies that use Turbo codes, such as LTE.

Existing PKC-based methodologies need to be more appropriate for resource-constrained WBAN, even though

traditional PKC necessitates a substantial portion of processing overhead. Because of the significant computational complexity, previous PKC-based approaches are unsuitable for resource-constrained WBAN. Many research projects have used "public-key cryptography (PKC)" to determine authentication methods [10] [11]. In [12], The author described a privacy authentication and group key management method for WBANs based on public key cryptography. However, this protocol lacked mutual recognition verification between the sensor network and the patient, making patient data conveniently intercepted. In this article, they propose a certificate-less authentication mechanism for safe authentication across untrusted connections. This study proposes a low-complexity security authentication system to accomplish mutual authentication between objects while guaranteeing that the user's confidentiality and data information is not leaked through an unsecured network [13].

Because the battery capacity of each IoT device is limited, it is best to reduce the energy consumed to extend the life of the medical system. This paper [14] describes deploying an IoT-based in-hospital healthcare platform model on the ZigBee mesh technology. The author's study describes numerous wireless media accessible for clinical situations, including one with a distinct function connected to mHealth and eHealth. NFC has been used to determine a person's current health state. Bluetooth Technology would be used to collect information from that very same room. LowPAN is used to track the condition of each unit in healthcare by using IPv6 equipment.

We could use a system that will easily demonstrate that almost all communication nodes are trustworthy to establish authentication in on-body sensor nodes. To establish sensor authentication, the researchers used sensors' accelerometer data to determine whether the devices are carried on the same individual's body around the waist. A method was proposed for evaluating walking patterns obtained by mobile accelerometers put in almost the same spot on the patient's body, and the findings showed that the patterns recorded by these sensors are alike [15].

According to Kirchhoff's' Principle, the security of a cryptosystem must be limited to the selection of its keys; all the others (along with the algorithm) should be deemed publicly available information [16]. This principal grabs that the crypto system should be secure, excluding the key, even though everyone knows everything.

This [17] article defines HMAC using a universal hash function. Specific HMAC new structures must declare a specific hash function. "SHA-1 [SHA], MD5 [MD5], and RIPEMD-128/160 [RIPEMD]" are current possibilities for such hash algorithms. Various HMAC versions will be designated by "HMAC-SHA1, HMAC-MD5, HMAC-RIPEMD", and so on.

The proposed scheme uses blockchain technology to ensure privacy-preserving and efficient authentication. Without the support of the RTA, one can build a blockchain platform and vehicles that use information to authenticate messages in a decentralized way using the blockchain architecture. The authors show how formal verification and implementation schemes can achieve several security goals and explain the overhead caused by blockchain procedures. Several difficult challenges face the blockchain-based vehicular network. They will recognize making a new consensus protocol for vehicles in the future to reduce infrastructure costs [18][19]. A scheme for efficient and secure group key agreements has been proposed. Symmetric encryption is used in conjunction with two secure hash functions. The registration phase in this scheme includes exchanging vehicle and RSU information by a trusted authority to each RSU and vehicle (TA). In the second phase, RSU authenticates the vehicle to form a group of vehicles. For message exchange, this vehicle group employs a group key. Encryption is used for authentication, while the hash function ensures message integrity [20].

TABLE I
COMPARISON OF RELATED WORK

Authors	Techniques	Limitation
Jaewon et al [18]	Blockchain-based MA using SHA-256 with PKI Mechanism.	Hashing protocols ensure data security but consume large computation and storage resources to manage
Maria almulhim et al [21]	Usage of ECC has been conducted for authentication with a comparison of Group Node and no Group node	More time to authenticate each node from individual nodes and make it vulnerable to delays to send data to the server.
Yu Yang et al [22]	CRC and KECCAK have been compared in the same environments.	Compared with parameters only with short size messages and area of implementation.
Kar J et al [23]	Combination of MD5 and SHA has been used for the IBOOS scheme	Repetitive usage of signing algorithms making it time-consuming

III. Preliminary Work

In IoT devices, energy is consumed at the application layer due to data rate, procedures, and programming models and manages processing and data procurement. Energy is consumed at the networking/routing layer when data link protocols and routing protocols join the system and Routes for packet forwarding are discovered [24].

The growth of the internet is based on IoT intended for the collection, analysis, and distribution of data via devices (WSN, WBAN) that build its essential module. A critical element of persistent IoT devices is their low resources. As we know low resource IoT devices are battery-powered so a traditional battery-operated device holds as its resources "storage, processing, bandwidth, and energy consumption" (Fig 5). As many pervasive IoT applications are limited in resources that is the reason to store, process, and share the data; numerous energy-efficient lightweight security protocols and algorithms are being deployed over the network [25]. IoT nodes are shown in Fig. 3.

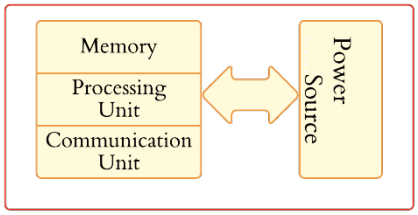

FIGURE 3. Typical IoT Nodes

A. Categories Of IoT Devices

IoT devices concerning hardware implementation and based on performance and competence can be categorized into 3 or mainly into 2 forms.

The 1st sort of IoT device is called High-end devices or High-power devices. These devices consist of single-board computers. Examples of high-end devices are Raspberry Pi, Panda Board, etc. smartphones or smart homes, etc. Because of their high level of resources, these devices are frequently utilized as IoT gateways, allowing them to handle new services such as providing intelligent analytics at the network's edge. High-power IoT devices are rich in resources and have satisfactory features to execute software based on OS like Linux [7].

The 2nd form of IoT devices is Low-resource/low-end or low-power devices. Our main focus of research also moves around these types of devices. Resources like energy, memory capacity, and CPU processing are limited in low-end IoT devices. Examples of low-end devices are Arduino and Tmote Sky etc. Low-end devices are further sub-categorized into 3 classes. Class 0, class 1, class 2[7] given in Table II.

TABLE I

CLASSES OF LOW-RESOURCE DEVICES

Stipulations	Class 0	Class 1	Class 2
Security exposures	Security exposures will have basic hazard	It can have medium hazard	It can create medium to high threat
Memory	<10kB	Approx. 10kB	Approx. 50kB
RTOS	No support	Could be implemented	Can be functioned
Communication Protocols	Gateways are used	Lightweight protocol i.e CoAP. Without gateways	HTTP
Flash	<100kB	Approx. 100kB	Approx. 250kB

When 3rd form is compared to low-end devices, which typically have clock frequency and random-access memory in the tens of MHz and KB, gadgets in this category frequently have clock frequency and RAM inside the hundreds of MHz and KB. Arduino Yun and Tessel 2 are two examples of middle-of-the-road devices [7].

B. Wireless Technologies

Traditional computing platforms gather data in the field but then just send it to a central data center where analysis is done, however, this is no longer a viable strategy. Wireless technologies may be divided into two categories based on the use case scenario: long-range wireless networking and short-range wireless networking. “LTE, LTE-Advanced, LTE-M, WiMAX, and, 5G wireless communication are used for long-distance wireless connectivity. Bluetooth and its variations, Zigbee, LoRa, NFC, RFID with EPC worldwide, and Wi-Fi “allow short-range wireless connection. The choice of a technology is determined by how well its qualities meet the needs of the use scenario [26].

IV. Security Requirements

Low-end IoT devices such as “meters, sensors, and Radio Frequency Identification (RFID)” tags often get inadequate or no encryption today, assuming that the information they extract is of little interest to intruders. Letting low-end devices vulnerable, on the other hand, may result in confidentiality breaches.

A. Confidentiality

To prevent data from disclosure, data secrecy is essential—many low-resource. IoT WBAN, and WSN devices are used in medical applications to communicate sensitive patient health information. An attacker can hear important information by listening in on the conversation. This eavesdropping may cause catastrophic harm to the patient because the adversary can use the started gathering information for several malicious activities. Encrypting the patient's data with a secret key communicated across a secure communication channel ensures confidentiality. Encrypting data via a secure connection ensures confidentiality [27].

B. Integrity

When communicated through an unsecured channel, an attacker might modify the message sent to the receiver, it is critical in the case of healthcare information. Due to a lack of integrity, the adversary can change the victim’s information before it reaches the BNC. This is extremely risky in the event of a life-threatening catastrophe. Proper data integrity procedures guarantee that the data acquired is not tampered with by an opponent. Message authentication mechanisms can help with this [28-33].

C. Availability

Availability indicates that the data and information should always be accessible when a service or a server needs them. This implies that the IoT devices used to detect the physical surroundings, the computer systems needed to store and interpret the data, and the communication links must all work efficiently. this is an important requirement in the case of healthcare systems; for example, the patient’s information must be available to the medical staff anytime they need it; an attacker can attack the info, and in life-saving applications, the patient can lose his/her life if the data is not available at the right time [29].

D. Authentication

Authentication can be of the message, device, and user authentication, and it is a very crucial part of all kinds of devices, especially in healthcare systems. It enables the verification that a trustworthy end device is transmitting the data. End devices generate a Message Authentication Code (MAC) for such data by exchanging a secret key, which informs the receiver that the data came from an authorized end system [9].

V. Analysis of Message Authentication Methods

Message Authentication ensures that the message was sent from a legitimate identity, not an imposter." Message authentication is commonly achieved using message authentication codes (MACs), authenticated encryption (AE), or digital signatures.

A. Message Authentication Code

The "Message authentication algorithm (MAC)" is a symmetric key encryption mechanism. The transmitter and receiver exchange a symmetric key "K" to create the MAC process. A MAC is effectively an encrypted checksum produced on the implicit message and delivered together with a message to ensure message validation. Figure 4 shows the generic model for message authentication using the symmetric key technique.

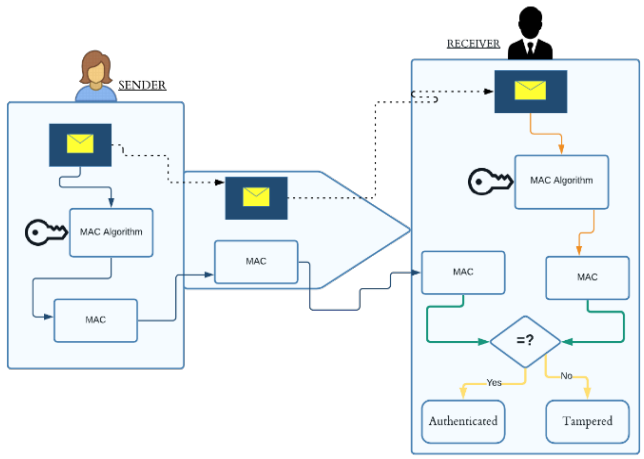

FIGURE 4. Message Authentication Using Symmetric Key

B. Digital Signatures

Digital signatures are based on PKC. It is a cryptographic value containing a message and a secret key only known by the sender or signer. A hash is created and that hash value and key produces the digital signatures. Signing algorithms i.e., RSA can be used. A digital certificate is an electronic document that a Certificate Authority issues (CA). It includes the key combo for a signature and the uniqueness affiliated with the key, such as the organization's name.

C. Keyed Hash-Based Message Authentication Protocol (HMAC)

HMAC is a block cipher code (MAC) generated by running a hash algorithm on the data to be verified using a confidential shared key. Like any other MAC, it is used for data integrity and confidentiality. The integrity of data must be checked for all transmission participants. HTTPS, SFTP, FTPS, and other communication protocols use HMAC. Cryptographically, hash functions such as MD-5, SHA-1, and SHA-256 can be used [17].

D. CRC

CRC is an error-checking method that adds a unique number to a set of techniques designed to check any modifications made during storage (or transmission).

E. AES

The Advanced Encryption Standard (AES) algorithm is a symmetric block cipher algorithm widely used worldwide.

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each

F. DES

The DES algorithm is a symmetric block cypher that provides adequate security while being inexpensive. Although the 56-bit key compromises security, brute-forcing this key takes several months and several computing engines. Even though DES has evolved into the advanced encryption standard (AES), many applications still rely on it for cryptography and data security.

G. MD5

MD5, or "message digest 5," is a popular cryptographic hash function. MD5 generates a 128-bit block size from the input data, typically expressed as a 32-digit hexadecimal number. MD5 hashes are distinct for different inputs, regardless of size.MD5 outputs 128 bits. That is, if we recognize all of the possible outputs of different inputs, we can have 2128 distinct outputs. As we can have a lot more than two inputs, we can definitely have a collision for 2128 distinct inputs.

H. SHA1

SHA-1 is a cryptosystem that uses an input to compute a 160-bit (20-byte) hash value. A message digest is what this hash value is called. A 40-digit hexadecimal number usually represents this message digest.

I. SHA2

It is an encryption algorithm developed by the National Security Agency of the United States to replace SH1. It generates a hash value of 224, 256, 384, or 512 bits. SHA2 certificates have been improved. While the hash produced by SHA2 is strong.

VI. Proposed Solution

A. Overview of Message Authentication

The "Message authentication algorithm (MAC)" is a symmetric key encryption mechanism. The transmitter and receiver exchange a symmetric key "K" to create the MAC process. A MAC is effectively an encrypted checksum produced on the implicit message and delivered together with a message to ensure message validation. The authentication is done by various message authentication protocols, which are heavy such as Hash functions like MD5, DES etc. But, in our proposed scheme, we have introduced a CRC-based lightweight message authentication protocol.

B. Diffie-Helman Bilinear Pairing

The DH key exchange protocol is not a full public-key cryptosystem; it only facilitates the transfer of a hidden value which could be used for symmetric keys or other reasons, but it does not help encryption or digital signatures. The Diffie-Hellman algorithm is the most commonly used in key exchange. Key exchange has always been challenging regardless of how fast and secure encryption algorithms are. You must devise a method of gaining access to all systems while utilizing the private key. The Diffie-Hellman algorithm facilitates this. The Diffie-Hellman algorithm would establish a secure communication channel, as shown in Fig. 5.

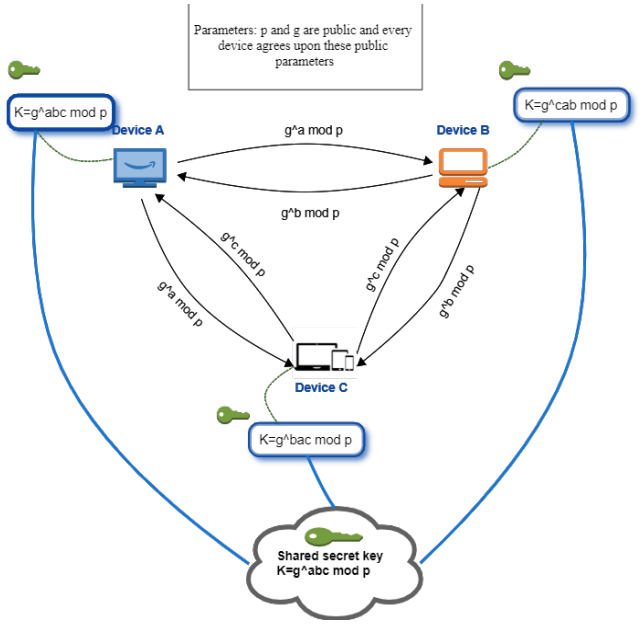

FIGURE 5. Working of Diffie-Hellman Between 3 Devices

C. Protocol Model

This proposed model will act as a message authentication protocol that will use CRC as a security primitive to send messages from one device to another device. Assuming there is an environment that has 2 devices, a client and a server as IoT devices. When a Sender sends a message to the receiver, the message is susceptible to numerous security breaches, such as any attacker manipulating or changing the message or reading it, compromising the security of communication. Our framework will ensure to authenticate the message securing it with CRC or Hashing protocol. The other feature we kept in mind is to make communication lightweight for devices such as in the medical field or military environment where it is difficult to change batteries regularly. To make it lightweight, we are using CRC to authenticate messages.

As we know, CRC is an error-detecting lightweight technique used to detect errors in digitally embedded networks. Firstly, we will link these users. In cryptography, we must share some secret key or keys to exchange messages. Our model uses a symmetric key exchange method by not exactly sharing but exchanging keys. To create a secure communication channel, the Diffie-Hellman algorithm will be used. After the keys have been shared, users can send messages to each other [Fig. 6].

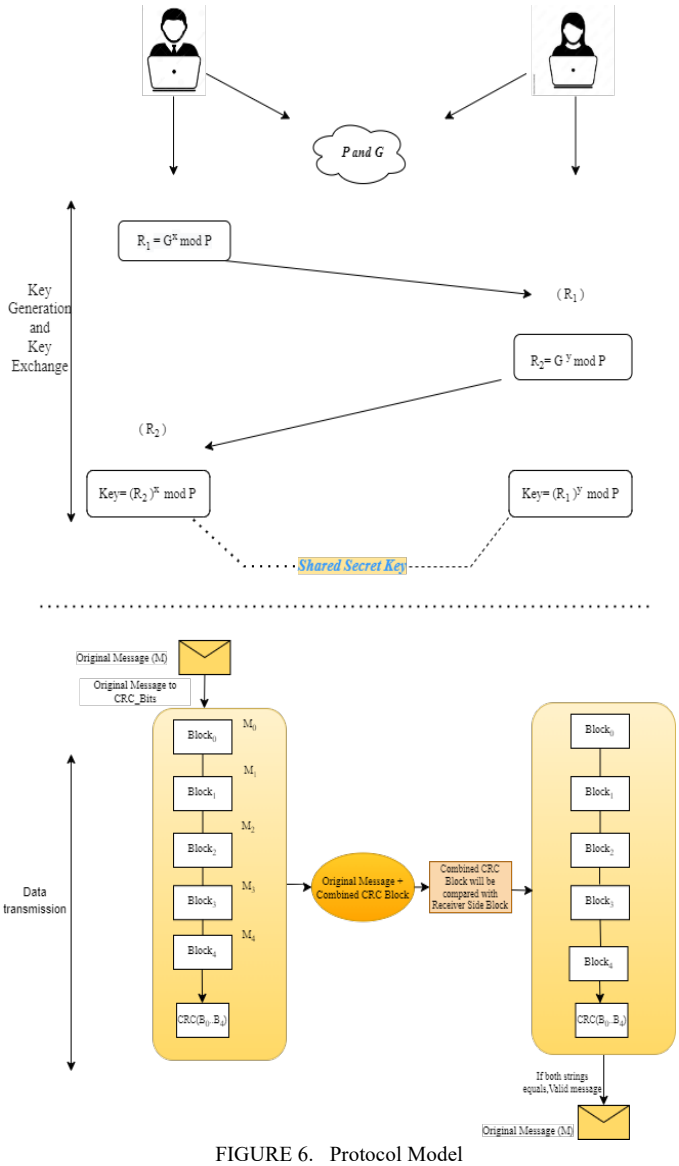

FIGURE 6. Protocol Model

Whenever User A(Sender) transmits a message to User B(Receiver), in order to make it more secure, the message will get divided into 8-bit blocks according to the size of the message. Each block will have the CRC itself, and if suitable it

will also contain the CRC of its previous block. After that to make it unique, it will be again combined in an array with CRC of blocks, CRC of that array will be taken and all these credentials along with the CRC polynomial key to DH will be sent to the user.

At the Receiver side, when the connection gets established, it will receive the original message, CRC polynomial, which DH used as key; the CRC of both sides will be compared as at sender side, if equal, then it will be considered valid, and the blocks will be valid after that and acknowledgment are sent to the sender that the message is valid. If an error bit is found, it will request for the retransmission of that error bit, and we don't have to resend the message blocks again.

VII. Implementation and Result

This section is crucial because it provides an overview of various aspects. This section presents our proposed security framework model and correlates CRC message authentication with hashing security protocol.

A. Message Authentication Using CRC Method

After the connection has been established and the keys have been exchanged, the user can send the message to the other user to which the keys are shared.

When sender A transmits a message or a file to receiver B, the message will be converted into binary bits, making it a 0 or 1 form. After the conversion is done, it will get divided into defined numbers i.e., 16 bits of blocks, and, the CRC of each message block is performed along with the key, making it a block of messages. After this, CRC blocks are passed on to a CRC Function and saved to the array list. The combined CRC of the whole array will be taken in which the CRC of each block is taken. After all these functions are performed, the original message, the Sender binary key (which in our case is the DH key), and the combined binary values after the CRC array is sent to the receiver side. In other words, the cipher text or encrypted bits are sent to the receiver [Fig. 7].

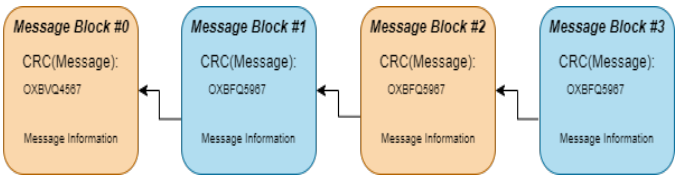

FIGURE 7. Abstract View of Message Blocks

It will Authenticate the message after combing CRCs and message bits we will take the CRC of the whole block and making it more secure for the receiver side. At the receiver end, the encrypted block is sent and the whole decryption will be performed as earlier. Then, blocks are again divided to convert it to the original message, if the combined CRC message from the sender side received is same as the combined CRC message generated at the receiver side, then the positive ACK is being sent to the sender that the message is valid and at Welcome message is shown at the Client side. If the bits of CRC message are changed, it will be considered as invalid message and a Negative ACK will be sent and it will be sent a retransmission request to the sender for only error bit blocks, otherwise it will be a valid message.

So, achieving message authentication and by using CRC it will be a lightweight message authentication and the result of this implementation will be discussed later in the next chapter.

B. Results

By using and implementing only CRC, we couldn't know the significance of CRC in Lightweight or low-constrained devices if we did not implement and compare CRC with heavy authentication protocols like SHA1, SHA2 and MD5. Although these protocols are more secure than that of CRC, sometimes some parameters must be compromised when utilizing one parameter, such as battery and memory consumption.

Following are the results we analyzed by taking into consideration parameters such as

- Battery consumption
- CPU usage consumption
- Time of Execution
- Memory Usage

The analysis has been made based on message size such as the variations in message characters or length. Multiple alterations have been made and every time the results change so we have noted down the nearest results possible (an average of 5 iterations). All these results are compiled with the standard of implementation and with the execution of code when the message is being sent from user A to User B, such as the Battery Consumption has been taken as a percent when not plugged in, CPU usage has been taken for the code execution, Time has been taken in Ms and Secs for small byte message and in min for large characters for execution, and memory usage has been taken in bytes [Table III].

TABLE II ALGORITHMS AT 22 CHARACTERS

Algorithms At 22 Characters	Battery Consumption	CPU usage Consumption	Time for Execution (sec)	Memory Usage in bytes
CRC	1	19.5	4.43	18011512
MD5	2	22.625	6.23	22323312
SHA-1	2	20.45	5.14	20136354
SHA-2	3	23.291	6.71	24116512

C. Finding

When the message to encode was of 22 characters, as mentioned above CRC took lesser time in Seconds and battery

consumption than that of all the other hashing algorithms.

CRC was faster than the MD5, SHA1, SHA2. SHA1 was faster than MD5 comparatively because of this reason the Md5 is outdated nowadays and SHA1 is more secure than the MD5.SHA1 and SHA2 are no doubt more secure than CRC and MD5 but as our intention is to make lightweight devices less consuming so as expected CRC consumed a much lesser battery consumption 3%, with a CPU consumption of 19.5% at 22 characters in 6 seconds with a memory consumption of 20 MB's [Fig. 8-11].

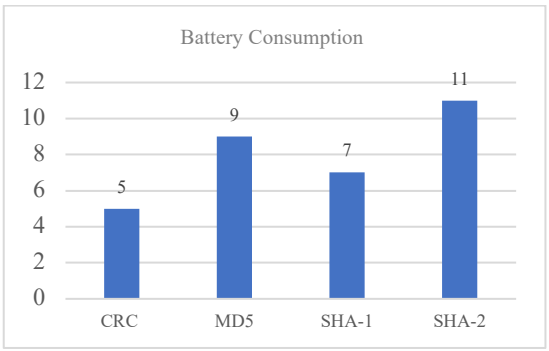

Figure 8 Protocol performance with respect to battery consumption

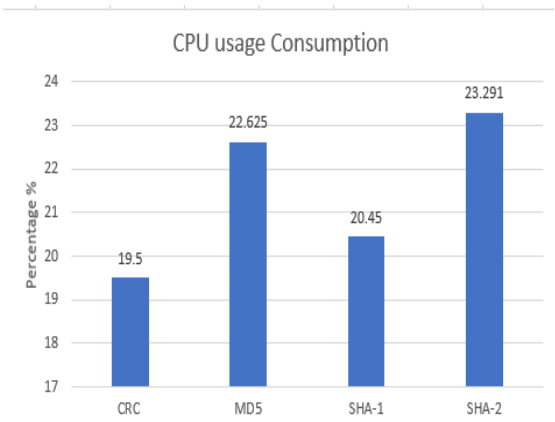

Figure 9 Protocol performance with respect to CPU consumption

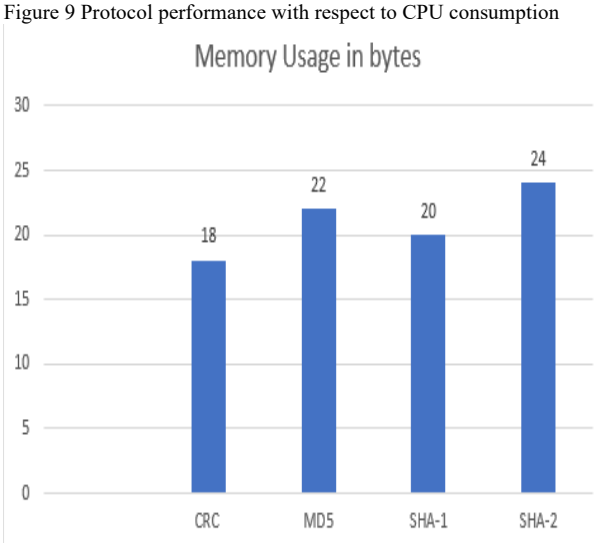

Figure 10 Protocol performance with respect to memory usage

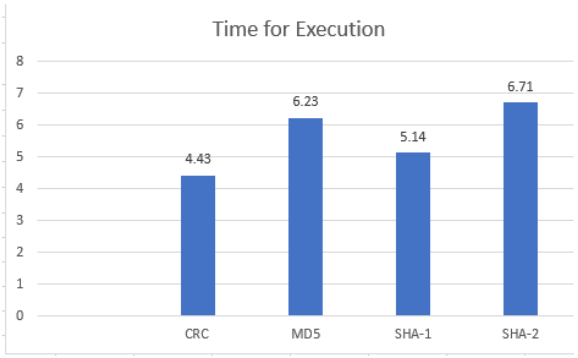

Figure 11 Protocol performance with respect to execution time

TABLE III ALGORITHMS AT 47 CHARACTERS

Algorithms at 47 Characters	Battery Consumption	CPU usage Consumption	Time for Execution(sec)	Memory Usage in bytes
CRC	2	20.1	33.52	20616511
MD5	6	27.33	115.44	27434612
SHA-1	4	22.99	80.59	24452421
SHA-2	6	27.78	120.31	29725411

D. Findings

When the message to encode was 47 characters [Table VI], as mentioned above CRC took less time and battery consumption than that of all the all the other hashing algorithms. CRC was faster than the MD5, SHA1, SHA2. SHA1 was faster than MD5 comparatively because of this reason the Md5 is outdated nowadays and SHA1 is more secure than the MD5.SHA1 and SHA2 are no doubt more secure than the MD5.SHA1 and SHA2 are no doubt more secure than CRC and MD5 but as our intention is to make lightweight devices less consuming as expected CRC consumed a much lesser battery consumption of 2%, with a CPU consumption of 20.1 at 47 characters in 13.24 with a memory consumption of 27 MB's [Fig. 12-15].

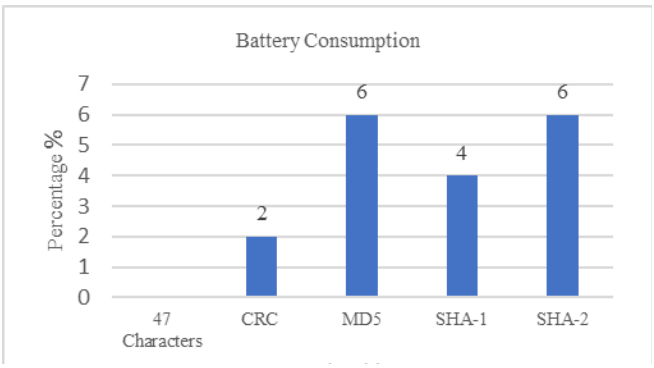

Figure 12 Comparison of CRC with other protocols

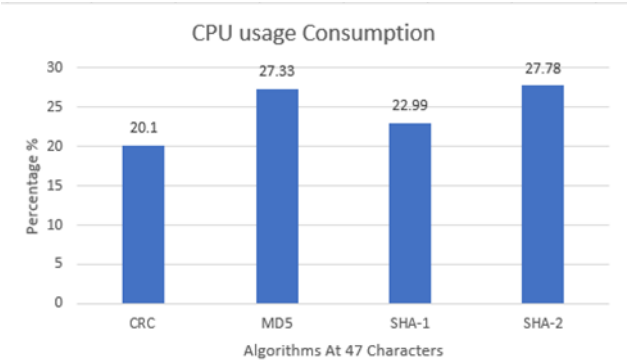

Figure 13 CPU usage-based comparison of CRC with other protocols

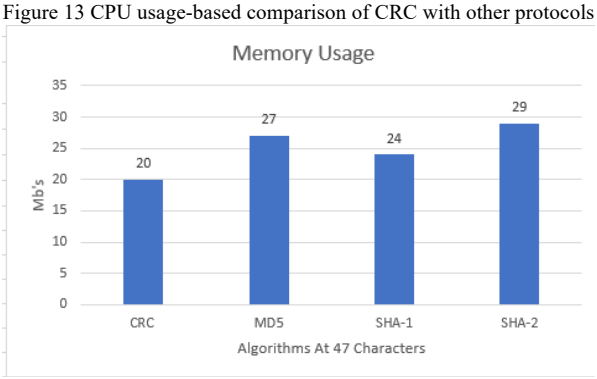

Figure 14 Memory usage based comparison of CRC with other protocols

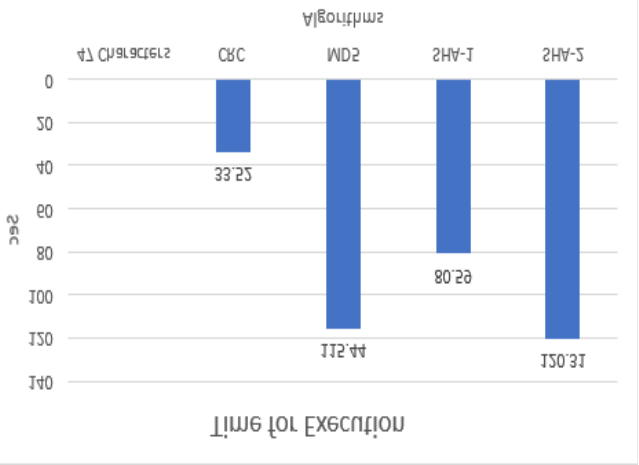

Figure 15 Execution time based PU usage based comparison of CRC with other protocols

TABLE V ALGORITHMS AT 88 CHARACTERS

Algorithms At 88 Characters	Battery Consumption	CPU usage Consumption	Time for Execution (min)	Memory Usage in bytes
CRC	5	22.45	1.44	22547651
MD5	10	29.254	3.11	29432145
SHA-1	7	24.56	2.54	27536354
SHA-2	12	30.11	4.48	31616512

E. *Findings*

When the message to encode was of 88 characters [Table V], as mentioned above CRC took lesser time and battery consumption than of all the all the other hashing algorithm.

CRC was faster than the MD5, SHA1, SHA2. SHA1 was faster than MD5 comparatively because of this reason the Md5 is outdated nowadays and SHA1 is more secure than the MD5.SHA1 and SHA2 are no doubt more secure than CRC and MD5 but as we intend to make lightweight devices less consuming so as expected CRC consumed a much lesser battery consumption 12%, with a CPU consumption of 22.45 at 88 characters in 15.44 with a memory consumption of 22 MB's [Fig. 16-19].

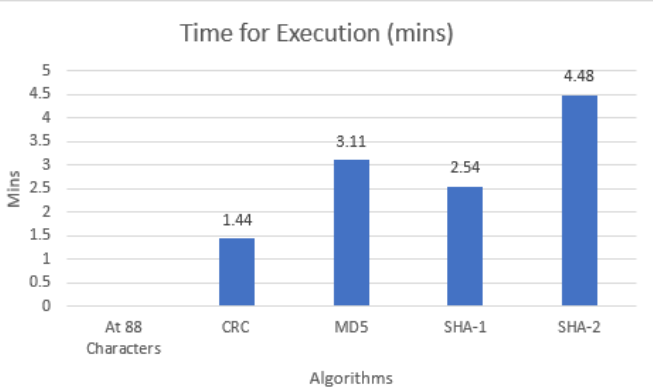

Figure 16 Battery consumption based comparison of CRC with other protocols

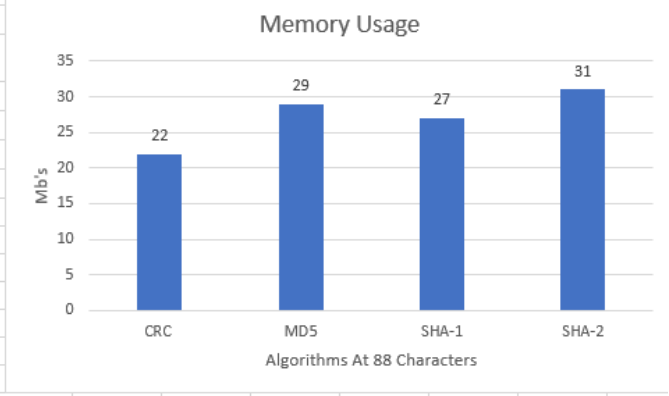

Figure 17 Memory usage based comparison of CRC with other protocols

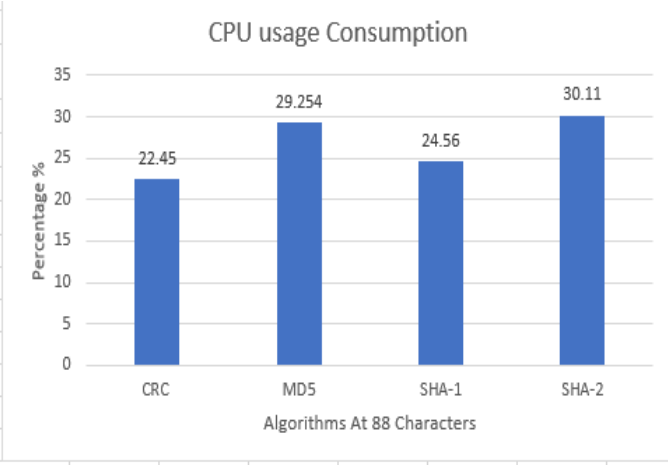

Figure 18 CPU usage based comparison of CRC with other protocols

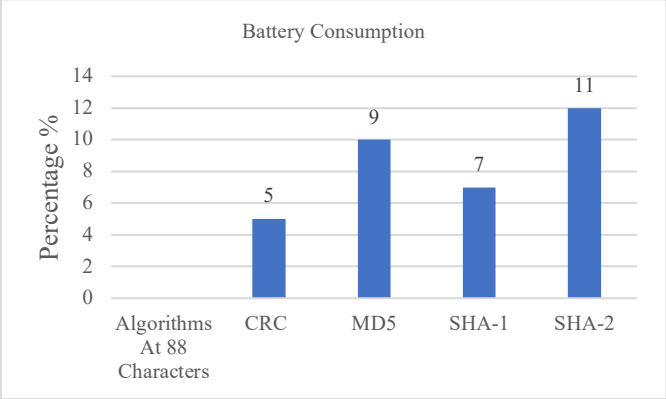

Figure 18 Battery usage based comparison of CRC with other protocols

VIII. Conclusion

This research aimed to develop a lightweight message authentication protocol based on CRC between two devices and compare it using other heavy hashing protocols like MD5, SHA1, SHA2. We examined the outcomes of both types of protocols and discovered that for low-resource devices, CRC obtained better results while exhibiting hashing protocols. Even though hash-based protocols are more secure than CRC-based protocols, CRC-based protocols work best on devices with limited resources and battery life. So, to obtain one benefit, another must be compromised. Whereas investigating these protocols, we concluded that CRC utilized protocol is faster than the other hashing-based protocols. We study these protocols by increasing and decreasing the message characters, with small message bytes 22 characters and with large message bytes 88 characters and the consumption of battery, memory usage, and CPU usage is also visibly less as compared to the hashing protocols.

IX. Future Work

The framework's success can be improved in the future by the additions I'm going to make. I am confident that with the knowledge I have gained from fostering this authentication framework, The protocol will be upgraded in the future by recommending device authentication for IoT devices to provide better security and performance with identity anonymity to enhance user experience. A fascinating challenge for the researchers would be to develop a more secure and effective model for less computation that uses the server or battery, as well as to find features of the IoT devices that are to be employed.

References

[1] K. K. Patel, S. M. Patel, and P. G. Scholar, "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges," *International Journal of Engineering Science and Computing*, vol. 1, pp. 6122–6131, May 2016, doi: 10.4010/2016.1482.

[2] R. M. Gomathi, G. H. S. Krishna, E. Brumancia, and Y. M. Dhas, "A Survey on IoT Technologies, Evolution and Architecture," in 2nd International Conference on Computer, Communication, and Signal Processing: Special Focus on Technology and Innovation for Smart Environment, ICCCSP 2018, Institute of Electrical and Electronics Engineers Inc., Aug. 2018. doi: 10.1109/ICCCSP.2018.8452820.

[3] R. Kandaswamy and D. Furlonger, "Blockchain-Based Transformation: A Gartner Trend Insight Report," A Gartner Trend Insight Report. Accessed: Dec. 30, 2022. [Online]. Available: https://www.gartner.com/en/doc/3869696-blockchain-based-transformation-a-gartner-trend-insight-report.

[4] N. Sharma, M. Shamkuwar, and I. Singh, "The history, present and future with iot," Intelligent Systems Reference Library, vol. 154, pp. 27–51, 2019, doi: 10.1007/978-3-030-04203-5_3.

[5] I. de Mendizábal, "IoT Communication Protocols—Network Protocols - Technical Articles," All About Circuits. Accessed: Dec. 30, 2022. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/internet-of-communication-communication-protocols-network-protocols/

[6] C. Worlu, A. Jamal Amri, and N. A. Mahiddin, "WIRELESS SENSOR NETWORKS, INTERNET OF THINGS, AND THEIR CHALLENGES 557," International Journal of Innovative Technology and Exploring Engineering y and Exploring Engineering (IJITEE) , vol. 8, no. 12S2, pp. 556–566, Oct. 2019, doi: 10.35940/ijitee.L1102.10812S219.

[7] M. O. Ojo, S. Giordano, G. Procissi, and I. N. Seitanidis, "A Review of Low-End, Middle-End, and High-End Iot Devices," IEEE Access, vol. 6, pp. 70528–70554, 2018, doi: 10.1109/ACCESS.2018.2879615.

[8] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," J Ambient Intell Humaniz Comput, vol. 1, pp. 1–18, May 2017, doi: 10.1007/S12652-017-0494-4.

[9] E. Dubrova, M. Näslund, and G. Selander, "CRC-based message authentication for 5G mobile technology," Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015, vol. 1, pp. 1186–1191, Dec. 2015, doi: 10.1109/TRUSTCOM.2015.503.

[10] D. He and D. Wang, "Robust Biometrics-Based Authentication Scheme for Multiserver Environment," IEEE Syst J, vol. 9, no. 3, pp. 816–823, Sep. 2015, doi: 10.1109/JSYST.2014.2301517.

[11] K. A. Shim, "Universal Forgery Attacks on Remote Authentication Schemes for Wireless Body Area Networks Based on Internet of Things," IEEE Internet Things J, vol. 6, no. 5, pp. 9211–9212, Oct. 2019, doi: 10.1109/JIOT.2019.2922701.

[12] S. L. Keoh, E. Lupu, and M. Sloman, "Securing body sensor networks: Sensor association and key management," in 7th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2009, TX: Enlighten PUBLICATIONS, Mar. 2009, pp. 1–6. doi: 10.1109/PERCOM.2009.4912756.

[13] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," Journal of Network and Computer Applications, vol. 106, pp. 117–123, Mar. 2018, doi: 10.1016/J.JNCA.2018.01.003.

[14] R. K. Kodali, G. Swamy, and B. Lakshmi, "An implementation of IoT for healthcare," 2015 IEEE Recent Advances in Intelligent Computational Systems, RAICS 2015, pp. 411–416, Jun. 2016, doi: 10.1109/RAICS.2015.7488451.

[15] N. Yenuganti, "Authentication in Wireless Body Area Networks (WBAN)," University of South Florida, 2016. Accessed: Dec. 29, 2022. [Online]. Available: https://digitalcommons.usf.edu/etd/6442.

[16] G. J. Simmons, "Symmetric and Asymmetric Encryption," ACM Computing Surveys (CSUR), vol. 11, no. 4, pp. 305–330, Dec. 1979, doi: 10.1145/356789.356793.

[17] Dr. H. Krawczyk, M. Bellare, and R. Canetti, "RFC 2104 - HMAC: Keyed-Hashing for Message Authentication," Yorktown, Feb. 1997. Accessed: Dec. 30, 2022. [Online]. Available: https://datatracker.ietf.org/doc/rfc2104/.

[18] J. Noh, S. Jeon, and S. Cho, "Distributed Blockchain-Based Message Authentication Scheme for Connected Vehicles," Electronics 2020, Vol. 9, Page 74, vol. 9, no. 1, p. 74, Jan. 2020, doi: 10.3390/ELECTRONICS9010074.

[19] L. Liu, Y. Wang, J. Zhang, and Q. Yang, "A Secure and Efficient Group Key Agreement Scheme for VANET," Sensors 2019, Vol. 19, Page 482, vol. 19, no. 3, p. 482, Jan. 2019, doi: 10.3390/S19030482.

[20] M. Wang, D. Liu, L. Zhu, Y. Xu, and F. Wang, "LESPP: lightweight and efficient strong privacy preserving authentication scheme for secure VANET communication," Computing, vol. 98, no. 7, pp. 685–708, Jul. 2016, doi: 10.1007/S00607-014-0393-X/METRICS.

[21] M. Almulhim, N. Islam, and N. Zaman, "A Lightweight and Secure Authentication Scheme for IoT Based E-Health Applications," IJCSNS

International Journal of Computer Science and Network Security, vol. 19, no. 1, 2019.
[22] Y. Yu, S. Tao, and E. Dubrova, “Comparison of CRC and KECCAK Based Message Authentication for Resource-Constrained Devices,” 2018 16th IEEE International New Circuits and Systems Conference, NEWCAS 2018, pp. 217–220, Dec. 2018, doi: 10.1109/NEWCAS.2018.8585692.
[23] Naseer, Fawad, Muhammad Nasir Khan, and Ali Altalbe. "Intelligent Time Delay Control of Telepresence Robots Using Novel Deep Reinforcement Learning Algorithm to Interact with Patients." Applied Sciences 13, no. 4, 2462, 2023.
[24] Naseer, Fawad, Muhammad Nasir Khan, and Ali Altalbe. "Telepresence Robot with DRL Assisted Delay Compensation in IoT-Enabled Sustainable Healthcare Environment." Sustainability 15, no. 4, 3585, 2023.
[25] Altalbe, Ali, Muhammad Nasir Khan, Muhammad Tahir, and Aamir Shahzad. "Orientation Control Design of a Telepresence Robot: An Experimental Verification in Healthcare System." Applied Sciences 13, no. 11, 6827, 2023.
[26] Khan, Muhammad Nasir, Syed K. Hasnain, and Mohsin Jamil. Digital Signal Processing: A Breadth-first Approach. Stylus Publishing, LLC, 2016.
[27] J. Kar, K. Naik, and T. Abdelkader, “A Secure and Lightweight Protocol for Message Authentication in Wireless Sensor Networks,” IEEE Syst J, vol. 15, no. 3, pp. 3808–3819, Aug. 2020, doi: 10.1109/JSYST.2020.3015424.
[28] N. Naji, M. R. Abid, N. Krami, and D. Benhaddou, “Energy-aware wireless sensor networks for smart buildings: A review,” Journal of Sensor and Actuator Networks, vol. 10, no. 4, Dec. 2021, doi: 10.3390/JSAN10040067.
[29] S. Zahoor and R. N. Mir, “Resource management in pervasive Internet of Things: A survey,” Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 8, pp. 921–935, Oct. 2021, doi: 10.1016/J.JKSUCI.2018.08.014.
[30] M. S. Hajar, M. O. Al-Kadri, and H. K. Kalutarage, “A survey on wireless body area networks: architecture, security challenges and research opportunities,” Comput Secur, vol. 104, May 2021, doi: 10.1016/j.cose.2021.102211.
[31] V. Sivaprasatham and J. Venkateswaran, “On the Security Issues in Wireless Body Area Networks,” International Journal of Digital Content Technology and its Applications, vol. 3, no. 3, pp. 1780–1787, 2009, doi: 10.3844/JCSSP.2012.1780.1787.
[32] H. Taleb, A. Nasser, G. Andrieux, N. Charara, and E. Motta Cruz, “Wireless technologies, medical applications and future challenges in WBAN: a survey,” Wireless Networks, vol. 27, no. 8, pp. 5271–5295, Nov. 2021, doi: 10.1007/S11276-021-02780-2/TABLES/3.
[33] P. I. Radoglou Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, “Securing the Internet of Things: Challenges, threats and solutions,” Internet of Things (Netherlands), vol. 5, pp. 41–70, Mar. 2019, doi: 10.1016/j.iot.2018.11.003.