A security mechanism for Enhanced ShockBurst wireless communication protocol using nRF24L01

Aref Ayati
Graduate University of Advanced Technology
Hamid Reza Naji (naji@kgut.ac.ir)
Graduate University of Advanced Technology

Research Article

Keywords: IoT, WSN, nRF24L01, Enhanced ShockBurst, Security, Moving Target Defense

Posted Date: December 25th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3777984/v1

Additional Declarations: No competing interests reported.

A security mechanism for Enhanced ShockBurst wireless communication protocol using nRF24L01

Aref Ayati, Hamid Reza Naji*

Department of Computer Engineering and Information Technology, Graduate University of Advanced Technology, Kerman, Iran

Abstract

The increasing use of the Internet of Things and Wireless Sensor Networks has been very noticeable due to the diversity of their applications. Due to the numerous cyber security threats and also the weaknesses of communication systems in IoT and WSN infrastructures, we have conducted research to increase the security of the "Enhanced ShockBurst" protocol which is one of the wireless network protocols that is used widely in these fields. We propose a security mechanism to enhance the security of the "Enhanced ShockBurst" wireless network protocol and protect communication networks in IoT or WSNs that use this protocol. This mechanism is more secure and faster than the previous proposed mechanisms and it is based on CIAA that can guarantee message confidentiality, integrity, availability, and accountability. In this method, by taking advantage of time and a function to frequently shuffle the address in the Enhanced Shock Burst wireless communication protocol, more suitable security conditions were implemented with a not-too-high cost compared to other methods.

Keywords: IoT, WSN, nRF24L01, Enhanced ShockBurst, Security, Moving Target Defense

1. Introduction

The expansion of the Internet of Things (IoT) and the wireless sensor networks (WSN) in recent years and the diversity of their applications, due to the fact that most of their nodes are connected to each other through a wireless network, is one of the main goals of the intruders to infiltrate these networks. In particular, IoT and WSN have been used in many fields such as business, automation, smart cities, agriculture, drones, healthcare, environment, and education. In such systems, often using a communication module that can communicate based on a communication protocol, nodes can exchange information in the network [1-4].

As communication technology advances and devices and computational systems become more accessible, the IoT is expected to expand rapidly. This makes IoT security a crucial issue to protect the hardware and the networks in the IoT system. However, since connecting these appliances is a relatively new concept, security has not been effectively a priority in the production of these appliances [5]. One of the network protocols that can help us connect each node in these networks to others wirelessly is known as "Enhanced ShockBurst" (ESB). Also, this protocol is used by the nRF24L01 communication chips family.

In recent years, we have seen the increasing use of nRF24L01 communication modules that operate based on the ESB protocol. Probably important reasons to use them are reasonable price, high efficiency, and high data transfer rate [6]. However, one of the downsides of this communication protocol is its security primitives. Nevertheless, the use of these modules has been observed recently in various industries such as the manufacturing of drones and devices related to the IoT [7,8]. In general, as it has stated in [9], the ESB

distributed network topology has now been developed without any default encryption or security process. As a result, the risks of cyber-attacks greatly threaten communications based on this protocol. Therefore, this protocol requires an effort to make it more secure in terms of security in the future.

In this paper, we will try to present a fast security mechanism that is somewhat complete in every aspect, based on previous research on communication security, so that in the future, users of the ESB communication protocol can make their communication more secure. The paper will present a security mechanism that ensures communication confidentiality, integrity, and user (node) authentication. Also, we present a mathematical relations-based address shuffling method for hardening and increasing the control of accessibility. The conventional methods of cyber defense (e.g., information encryption, authentication, vulnerability scanning, and virus protection) have offered some level of security, but they are not enough to defend against evolving and diverse attacks, and these attacks require more advanced cyber defense techniques [10,11].

As we know in the field of computer networks, if communication is made, the details in the network remain the same until the end of communication. In this case, the intruder can easily observe and eavesdrop on the network, analyze it, and take action against it [10]. Also, some cyber-attacks such as Denial of Service (DoS) or SYN Flood, etc. are also likely to happen anywhere and anytime in a traditional network [12]. That is why we use the address shuffling method which is a variant of the Moving Target Defense method known as MTD.

Imagine that we have a sensor node or a node in an IoT network that needs to send a message to another node. We need to make sure that our message is safe from any kind of eavesdropping which is a reason to have a message confidentiality mechanism. In addition, all messages must be sent in an integrity manner, and we must also ensure the guarantee of this process that no message has been tampered with, and at the end, have a simple node authentication to defend against all types of impersonation attacks for accountability.

In this paper first, we will present other works in this field as related works, in continuing, we have a discussion about the ESB communication protocol, and then we propose our security mechanism. We have implemented our proposed mechanism on a microcontroller to show its performance and compare it with other methods. Finally, we have the conclusion and future works.

2. Related Works

As an example of ESB applications, we can refer to some studies in this field. In [13], while stating that these days the most important challenge in the agricultural industry and in the direction of the digitization of agriculture is improving productivity and profitability, authors addressed irrigation management and used the IoT in this direction. In this project, the nRF24l01 module is used for radio communication between the sink nodes and the other nodes, but there is not any specific security mechanism to secure nRF24l01 communications. On the other hand, in [14] again we see a discussion about providing an agricultural system based on the IoT, and ESB-based nRF24L01 modules are used, which again does not include any security mechanism.

In other research [15], the IoT has been used with the nRF24L01 wireless communication module in the field of nutrition. However, there is no security mechanism to establish secure communication.

In other health-related cases [16], for better maintenance of the elderly or other vulnerable groups with mental problems or Alzheimer's, etc., the use of the IoT with wireless communication through nRF24L01 has been discussed but they didn't provide any security mechanism. In some other works, security mechanisms to establish minimum security in nRF24L01-based communications have been discussed, which we will review them in below.

In article [17], by using an MSP430F1611 microcontroller manufactured by Texas Instruments (TI) and a communication module nRF24L01 based on ESB, they tried to establish a secure communication by the microcontroller. they compared their new design with MSP430F1611 in terms of energy consumption. This article, presented in the field of health, notes that most of the articles in recent years, especially in the field of WSN, have only tried to present and simulate a proposed method using symmetric encryption, but in practice, they have not been able to implement it in a real environment. CCM security mode, which is one of the types of encryption modes with the AES-128 algorithm, is used to establish the confidentiality of the data and also ensure the integrity of the message. In this method, in five working modes using the CCM method and AES-128 encryption, a Message authentication code (MAC) is first applied to the message, then the combined message and MAC are encrypted and transmitted on the communication medium. These five working modes are specified in two types of message frames, and in the first and third modes, the message frame contains a combined 4-byte MAC with an 8-byte original message in the payload part of the message frame. In the first mode, the hello message is sent from the sensor node to the master node to enter the network, and in the third mode, it is used to send the ACK from the sensor node to the master node or vice versa. Three other working modes are also used to transmit normal messages between the sensor nodes and the master node, where a combined 4-byte MAC with 24-byte of original message text is used in the payload part of the message frame. As mentioned, all messages are wholly encrypted with the AES-128 method before being sent to the network. In this method, no special provision has been made to prevent all types of DoS attacks. The system proposed by these authors is generally vulnerable to all types of DoS attacks due to the lack of any defense mechanism against it, on the other hand, due to the technique of first MAC setting and then encryption, it is also can resist tampering attacks. But it is vulnerable in a way, and if the intruder changes the content after listening to the network, the receiver of the message, as long as he does not decrypt the message, does not know about the content change. Another notable thing is the absence of any authentication mechanism, which ultimately makes this system vulnerable to authentication-based attacks.

The researchers of the article [18] also implemented an asymmetric encryption algorithm on the communication between two nRF24L01 modules based on the ESB network protocol to encrypt all data transmitted on the network. They used the RSA encryption algorithm simply because according to their research, compared to the RC5 and Skipjack algorithms, the RSA algorithm performed better, as in general asymmetric algorithms provide more security, but they did not consider the energy consumption. Also, they stated this algorithm enabled them to be safe from eavesdropping attacks and they established authentication and security only with this type of encryption. It seems that the focus of this research was on maintaining the confidentiality of data but not on other security factors. Due to the absence of any message authentication, intruders can disrupt the integrity of the messages, which leads to tampering attacks.

In another study, the authors of the article [19], have been trying to provide and implement a communication and data security plan on the infrastructure based on smart toys in the field of the IoT. They introduced a method to secure communications based on ESB protocol. By presenting some scenarios, they intended to provide a mechanism for securing communication in the infrastructure of smart toys. They intended to connect a smart toy to the data collector using the nRF24L01 module and transfer information from the toy to the data collector. They transfer the information to the Android cellphone via Wi-Fi communication protocol and then to a server called EDUCERE using the Internet. In the initial part of the communication where nRF24L01 is used for the connection between the toy and the collector, the researchers in this project investigated two security threats and provided a solution to overcome these two threats. The first threat is that if a fake collector sends the wrong messages to the toy, the service of the toy will suffer. This case was considered as a DoS attack and suggested that if the MAC mechanism is used in the text of the transmitted message and then the MAC is checked for each message, the DoS attack problem will no longer occur. But in our opinion, the MAC mechanism does not provide security against all types of DoS attacks, and these methods create security against attacks such as tampering attacks. Also, the goal was to prevent sending wrong messages, but since they created the MAC digest for their original message and then tried to encrypt a combination of them, this will cause the method to spend a lot of energy to decrypt and then check the MAC for tampered messages. The second threat they addressed is a fake collector steals messages and creates a problem for confidentiality in the network. They have not mentioned exactly what attack they are protecting against, but it seems that they intend to establish security against man-in-the-middle (MITM) attacks and message eavesdropping attacks. To solve this problem, their suggestion was to use the AES-128 encryption algorithm along with a MAC to ensure confidentiality and authenticate message integrity. However, they did not specifically mention why they used AES-128 and simply stated that due to the limitation of processing resources in smart toys, they turned to this symmetric encryption method and ignored other methods. The main point in the method used by them is the mechanism of transferring the session key on the network which in case the primary key is leaked, which also can be by a malicious insider as well. The session key can leak by eavesdropping through a message transmitted on the network. They have used the MAC setting with SHA-3 and the Keccak methods using multiple XOR operations on variables for making keys and encryption by AES-128. This method is vulnerable to DoS attacks, but due to the use of the third version of SHA's MACs, it can prevent tampering attacks, but a specific mechanism for authenticating nodes is not visible.

In [20], researchers developed an intelligent street light system using cellphone software, in the meantime, they entrusted the communication between the sensor nodes to the nRF24L01 module and worked on some of its security issues. In general, their model is based on an Android application on a cellphone that is paired with a master node in a WSN network, which is synchronized with other common nodes that are each paired on a streetlight. In the method considered by these authors, the Payload part which was in each ESB message frame is filled with just seven bytes from 32-byte and to establish security, they only considered the confidentiality and used an RC4 encryption algorithm with a key length of 128 bits which is a stream cipher encryption algorithm and not a block cipher. All seven-byte message content is encrypted by this symmetric encryption algorithm and decrypted on the destination device. According to the authors, the reason for using the RC4 algorithm is the faster operation speed and less memory consumption of stream

ciphers compared to block ciphers. The researchers in this article, of course, suggested to use of cryptographic algorithms such as RSA, DES, and AES for more efficiency in security. It should be noted that due to the inherent weaknesses, Microsoft ended its support for this algorithm in its web browsers in 2015, and the reason was the capacity of this algorithm to restore encrypted texts [21]. In this method, there isn't node authentication and message authentication, which consequence of attacks such as identity change, password guessing along with Sybil attacks, and on the other hand, the lack of MAC also causes vulnerability to data tampering, also it does not have any mechanism to resist against DoS attacks.

Almost all of the discussed related works did not consider node authentication in the ESB communication protocol and they didn't have a mechanism to defend against password guessing, impersonation, and Sybil attacks based on accountability requirement.

3. The Enhanced ShockBurst Protocol

This protocol transfers data based on the packet frame described in Fig 1 This protocol was developed by the Nordic Semiconductor company and has a low-consumption, low-delay, low-cost, high-speed transfer rate. While it provides reliable two-way communication, is suitable for communicating between devices such as those devices that work based on WSNs and the IoT infrastructures. This protocol, like some other wireless network protocols, is capable of supporting the authentication mechanism, encryption, and message authentication code [22-24].

Preamble (1 Byte)	Address (3-5 Bytes)	Packet Control Field (PCF) (9 bits)	Payload (0-32 Bytes)	CRC (1-2 Bytes)

Fig 1 Packet Frame of Enhanced ShockBurst Protocol.

Each packet frame of the ESB protocol consists of 5 main parts: 1-Preamble 2-Address 3-Packet Control Field (PCF) 4-Payload and 5-CRC.

At the beginning of the message frame, we see the "Preamble", which is a sequence of 01010101 or 10101010. The "address" section, contains the address of the module in receiver mode. The address of the receiver in the transceiver module can be set with 3, 4, or 5 bytes. The next part in the message frame is called "PCF" as you see in Fig 2 which contains three components. The first one represents the "length of the payload" of the sent message (content). The next one is "PID" to find out whether the received packet is new or duplicate. The use of this part in the original message packet will prevent duplicate data from being sent to the main microcontroller in the network node, and in fact, it will be somewhat resistant to replay attacks. In the following, the "ACK" part is used to confirm the message receipt. Outside of the PCF part, we also reach the main part of the main content of the message "Payload", which can contain up to 32 bytes, and at the end, "CRC" is used as an error detection mechanism in the entire packet, which is 1 or 2 bytes and it is calculated based on the Address, PCF and Payload fields.

Payload Length (6 bits)	PID (2 bits)	No_ACK (1 bit)

Fig 2 Details of PCF.

As can be seen in the general message frame of this protocol, there is no trace of a special security mechanism. In the following, we suggest making changes in the address and the payload parts to make this protocol more secure against some cyber-attacks.

Currently, unreliable communication channels and unattended operations have made the security defense processes more complicated [25], which happens to be as mentioned in the ESB protocol, this protocol provides us with an unreliable communication channel in the basic state. In this paper, we address some passive and active attacks that can analyze or eavesdrop on our communication or can tamper or modify our packets [26]. In the next parts of this article, we will discuss our proposed mechanism to create a secure channel in ESB-based communications that provides security against passive and active attacks. This secure channel will guarantee the confidentiality, authentication, and integrity of the message based on [27] theory, and also try to have a lightweight authentication to guarantee accountability.

Based on [5] and [28] currently, some threats for IoTs or WSNs are eavesdropping, man in the middle, Sybil, impersonation or identity spoofing, DoS, password guessing, tampering, and SYN flood attacks. In this paper, we suggest a mechanism for addressing the above issues. The ESB communication protocol is currently working with the nRF24L01 module family which can provide up to 2mbps data transfer rate and long-range communication around over 1 km if you use new versions of nRF modules with an amplifier.

4. Defaults of The Proposed Security Mechanism

A. Security Primitives of the Proposed Security Mechanism

We consider security and efficiency in designing our system as follows.

Security: Establishing security in communication networks is done to prevent the occurrence of active or passive attacks. Attacks that lead to eavesdropping or monitoring of the network or things like that are called passive attacks, attacks that cause data tampering or create a wrong flow, and things like that are also called active attacks [26]. In general, the four main factors to establish security in our proposed mechanism will be based on confidentiality, integrity, availability, and accountability called CIAA requirements [29]. To establish security in a network of IoT or WSNs, user or node accountability and confidentiality of messages must be considered, and the messages sent must be received authentically at the receiver's side, so we must have a solution to guarantee message integrity. We should also plan so that if a node is stolen or captured from the network, the intruder cannot access the information of the node or extract the information from the node, and we also need to plan to resist Denial of Service (DoS) attacks.

Efficiency: Due to the limitation in the resources of a node in the IoT or WSN, especially the limitation in the energy resources, the presented method should be accompanied by the maximum speed and the minimum consumption of energy and resources. Based on this, we will focus on choosing fast and low-power consumption algorithms.

B. Prerequisites

Since IoT or WSN systems can be viewed as distributed systems, cryptographic algorithms are widely used to ensure privacy in distributed systems [30]. Due to the limited resources and energy consumption consideration, and also providing encryption/decryption speed in symmetric encryption algorithms compared to asymmetric ones, we proposed our method based on the symmetric [31,32].

Symmetric Encryption Method: Symmetric encryption methods are used due to ease of use, simplicity in implementation, high execution speed, more flexibility, less memory consumption, and less power consumption. Among the symmetric encryption algorithms,

AES, Blowfish, and RC4 currently have good conditions in terms of algorithm execution speed on a message. We chose AES to benefit from the above issues and its high public acceptance and security [33,34].

User/Node Authentication: In user/node authentication, usually, both sides of the connection are authenticated with a username or password or a secret value. Due to the limitation in processing resources and the limitation in payload length in ESB communication Protocol, we refrain from providing a complex authentication method and settle for a very lightweight method, but we consider user/node authentication for each message exchanged in the network for to guarantee the user/node accountability.

Message Authentication or Integrity Check: Message authentication, is usually to confirm that the message was sent by the intended sender, that its content remained unchanged, and that it was sent at a specific time. Methods such as HMAC, LMAC, etc. are usually used for this purpose.

Generally, in these methods, a value called MAC is created by the sender using a specific algorithm and a shared key between the receiver and the sender, placed next to the message body and sent over the network. On the receiver's side, the main part of the message is separated from the MAC part and the MAC is produced again with the same key and algorithm using the original message. Next, the new MAC value is compared with the MAC value sent by the sender, and if they are equal, the message is accepted in the system. Otherwise, it means that the message has been tampered with by a possible adversary or intruder along the communication [30]. The MAC digest is used behind or in front of the message, which is generated using different algorithms. Currently, MD5 and SHA-1 algorithms have been generally retired by NIST [35], Also, MD4, GOST, HAVAL-128, and RIPEMD algorithms have been scientifically reviewed and rejected in terms of security [36].

Currently, SHA-2 and SHA-3 family algorithms are very popular and widely used, but unfortunately, the length of the digest produced by them is often very big, which is at least 28 bytes. They can be used but we need to use a truncate function to reduce the length of them which can force the system to do additional processes.

In our proposed mechanism, the SipHash algorithm will be used for generating MAC digest, which has a shorter digest length and a very suitable calculation speed. This algorithm was presented in 2012 and the digest output is equivalent to 8 bytes [36,37].

MTD in Dynamic Defense Method: In the past years, common solutions such as access control, information encryption, intrusion prevention systems, firewalls, anti-virus systems, etc have been provided. They provide a suitable level of security, but mostly they can provide security against attacks that are already known [38,12,10].

In the following, we have some discussions under the title of dynamic defense, which, with continuous changes in the network conditions, tries to deceive the attacker and increase his costs to discover a way of penetration. Also, it forces the attacker to waste a lot of time to penetrate the system. In this regard, we have selected a dynamic defense method based on a technique known as moving target defense (MTD) to implement in our security framework [39].

In the MTD technique, an attempt is made to constantly change the attack levels to make it harder for the adversary or intruder to recognize the target system, which wastes his time and increases the complexity of his work. By using this technique, the adversary can hardly find and use a precise method to act against the target system. In the MTD technique, active changes such as network address

change are done continuously and dynamically. The use of this technique in the methods of establishing security in the ESB communication protocol can also be a different, efficient, and effective innovation. In our proposed mechanism, one of the methods that we tend to use is the topic of changing or shuffling the relationships between addresses and devices in the communication network known as address shuffling.

In the so-called method of shuffling addresses, there are two main patterns for work [40] which we explain in the following:

- *Hopping pattern:* This pattern is accurate in terms of communication and time synchronization and is suitable for communication that is currently transferring basic data. The working method is based on the fact that both parties are well aware of the information of the hopping pattern of their addresses or one side of the communication is well aware of the hopping pattern of the other side. The synchronization mechanism in this method is done either by determining the time by exchanging the hopping pattern between the parties, or by using a preset function with an initial input value.
- *Mutation pattern:* In this pattern, communication synchronization is not time-accurate, and synchronization is usually supported by routing updates and DNS request/response or other third-party mechanisms.

Due to the use of the ESB wireless network protocol and the importance of accurate time synchronization and the reluctance to transfer a new address on the network, we will use the hopping pattern, in the proposed method of our security mechanism.

5. The Proposed Mechanism

In our proposed method, a continuous address shuffling mechanism, lightweight user/node authentication, integrity check, and message confidentiality will be used, which we discuss separately below:

1) Establishing address shuffling against DoS attacks for ESB communication protocol

In the proposed security mechanism to increase the security of the ESB protocol, our priority is providing a method based on MTD. To deal with attacks resulting from network analysis (such as DoS attacks), these attacks can be prevented by shuffling continuously the address based on the hopping pattern that was mentioned earlier. In fact, by creating conditions for the attacker to waste time, we reduce the possibility of him exploiting the vulnerabilities of our system, and even in a longer view, we can think that the attacker will also lose the possibility of scanning for vulnerabilities. More precisely, with continuous address shuffling, the attacker cannot easily track the target address and has to scan a larger address space to find it. As mentioned before, to establish communication in the network consisting of ESB protocol, it is necessary to specify an address for senders and receivers. We know the address length of the ESB protocol is between 3 to 5 bytes. In our proposed method, we consider the time variable (in minutes) along with an initial address that plays a role similar to the Initial Vector. The two mentioned components are combined in a mathematical process in which we shuffle the address and create a new address. We consider a normal mathematical function that has a limited computational load, and to create chaos in its output so that it is not possible to simply guess the entire function or form its sequence, we use a function called Mobius.

Before providing other definitions, we need to explain abbreviations in Table 1.

Table 1 Abbreviations table.

Abbreviations	Descriptions
PA	Pre-Address
CA	Current Address
CTM	Current Time Minute
NA	New Address

Definition 1: We define the Mobius function or $\boldsymbol{\mu(n)}$ for each input of $\boldsymbol{n} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}, \ldots\}$ with three outputs $\{\mathbf{1}, \mathbf{0}, -\mathbf{1}\}$ in such a way that we consider $\boldsymbol{\mu(n)}$ as the nth roots of the first unit. The function has the following three conditions:

- $\boldsymbol{\mu(n)} = +\mathbf{1}$ if $\boldsymbol{n}$ is a positive square-free integer with an even number up to the first factor.
- $\boldsymbol{\mu(n)} = \mathbf{0}$ if $\boldsymbol{n}$ has a first square factor.
- $\boldsymbol{\mu(n)} = -\mathbf{1}$ if $\boldsymbol{n}$ is a square less positive integer with the odd number up to the prime factor.

An example of the outputs of the mentioned function is in shown in Fig 3.

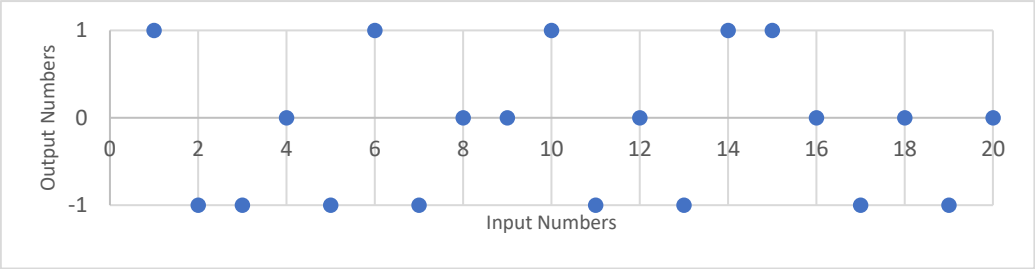

Fig 3 Mobius Function graph.

and in the following, using the Mobius function, we will generate a new address every three minutes.

Definition 2: $\boldsymbol{PA} = ([(\mathbf{9.8} * \boldsymbol{CA})] + \boldsymbol{CTM}\%\mathbf{5})$.

Definition 3: $\boldsymbol{M} = \boldsymbol{\mu}(\mathbf{PA})$.

Definition 4: $\boldsymbol{NA} = \boldsymbol{PA} + |\boldsymbol{M}|$.

In the above functions, the CA of the device is used as the Initial Vector. Before starting the operation, we need to set a starting address for the ESB protocol and we call it CA. Then, we calculated the product of CA in 9/8. The reason to use 9/8 is to simultaneously increase the value but not to exceed the extracted space for the address space as we have a space limit based on the payload of the packet frame. It is simply an attempt to implement a mechanism to continuously shuffle the address used in the ESB protocol. Next, by setting the Mobius function, we try to create chaos in the behavior of our function. We calculate the absolute value of the result of the Mobius function to make the guess of numbers more complicated and add it to the PA to create a new address. This process takes place every three minutes. Choosing the numbers three and five in this mechanism is just because we need two different numbers. On the other hand, instead of the number three, we can choose fewer numbers to shuffle the address faster. In this method, we are simply trying to ensure the possibility of frequent address shuffles for ESB protocol. Previously, however, in this case, a similar activity had not been seen, but in the field of computer networks, trigonometric, exponential, and logarithmic functions have been used a lot to realize the shuffle of addresses with the hopping pattern.

To prove the effectiveness of this method, we must conclude that the adversary faces a challenge in this type of attack (types of network analysis attacks, DoS, SYN Flood, etc.).

As a sample scenario, consider a simple network consisting of a transmitter node T and a receiver node R. Using the mentioned address shuffle method, the address of R is shuffled every 3 minutes. Suppose that the first address of R is 200, then based on the NA determination formula we obtain Table 2.

Table 2 Examples of generated addresses.

Addresses	Minutes
200	**0**
225	**3**
256	**6**
289	**9**

At first, we have address 200 as the first address, then after 3 minutes, the address will be 255, and also the rest of the addresses will be changed. Suppose the intruder C wants to use a SYN flood attack or any kind of DoS attack to involve the resources of the R system. C must calculate R's address at the same time in each period and then send SYN packets to it, which is usually very expensive for him. C should calculate NAs and update its packets by scanning and analyzing the network and its traffic. At the same time, R is planning to enter the NA. It is good for C to wait and fully analyze the network and form a sequence of addresses, but it faces problems in forming the common ratio of the function. On the other hand, it may even have a problem in discovering the primary address. For example, if we assume that C parses 50 addresses per minute, it will only take 4 minutes to reach our initial address. Now, as a mathematical problem, we know that the Mobius function is a function of number theory that we used in shuffling the address. Node R calculates its new address every three minutes. We know that DoS attacks of any kind are based on static addresses.

Theorem 1: Mobius function has a random property. That is, for every positive integer n, the probability that $\mu(n)$ is equal to one of the values 1, 0, or -1 is not equal.

Theorem 2: Mobius function is an asymmetric mathematical function and for both numbers m and n, the probability $\mu(m) = \mu(n)$ is very low.

Theorem 3: Mobius function has discontinuity property and for each n, the probability of $\mu(n) = \mu(n+1)$ is very low.

With the above three theorems, it can be said that the Mobius function causes the NAs of R to shuffle randomly, asymmetrically, and discontinuously in each period, and this means that the adversary C cannot find a specific pattern for shuffling addresses using prediction or regression algorithms. As a result, DoS attacks against this method become ineffective and much harder.

The next basic issue to create synchronism on both sides of the communication is to have the same clock. Our suggestion to increase the accuracy is to get the exact time value from a third factor such as a Global Navigation Satellite System (GNSS) or GPS to get the exact CT. An important point is that in GPS protocols, due to their message frame, they provide us with the time in raw and separated form, we just used the minute value of the hour to make it more complicated to guess our addresses. In this case, both parties will be aware of each other's address shuffle mechanism, without transmitting each other's current or future address on the communication network, which increases the security factor.

If this mechanism is established to maintain availability, the adversary will face a serious challenge for most attacks, especially DoS attacks. we think MTD methods similar to our method, can be used in UAVs, drones, and critical IoT or WSNs systems because such methods are very simple and lightweight and good for limited resources.

2) Establishing accountability with user/node authentication against identity change, Sybil, and impersonation attacks

In the following, to complete the proposed mechanism is lightweight, less expensive, and suitable for systems based on the IoT or WSNs with limited resources is using a secure secret value (SSV) as a password that both parties of communication or even other parties in a distributed system are aware of it. To iteratively authenticate the parties, the emphasis in this method is on iterative and continuous authentication in each message. Note that this amount of the SSV should be granted to the network nodes by a trusted person or group because if this is set by normal users of this system most people tend to choose a simple password to avoid remembering complex passwords [41,42]. On the other hand, one day a person may appear as a malicious insider, which ultimately leads to impersonation attacks. Our suggestion to generate this secure secret value is to use the ASCII template, where 95 characters are present and can be used to create this SSV. These 95 characters include 52 upper- and lower-case English letters, 10 numbers, and 33 symbols. For increasing the complexity of fast password guessing even with a computer, we suggest using an SSV of 8 bytes to maintain the minimum level of security. For example, a suitable SSV could be the value " *a$4D2s6 ". The message packets sent from each sender to each receiver consist of the content in which the information of the SSV is included first, and then the original message as mentioned in Fig 4.

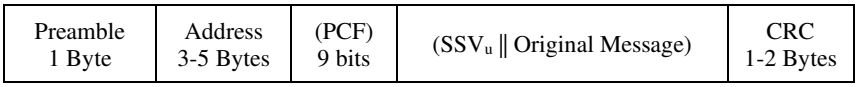

Fig 4 Message frame with a lightweight authentication.

In case of sending such a message on the destination side, the receiver must first separate the two main parts of the message and the SSV, and check it so that if it is correct, the original part of the message is considered.

3) Establishing the message integrity with message authentication code (MAC) against data tampering attacks and confidentiality with encryption against eavesdropping and MITM attacks

To establish the integrity of the messages, we suggest using the SipHash method. The reason for this choice is the suitable speed, low complexity, and proper security of the SipHash based on comparative tables. Most importantly, some other available options, such as MD5 or SHA1, which were very popular and did not have a large message digest length, are obsolete. Some others, have a large message digest output that will be a challenge for us to fit in the payload part of the ESB message frame, and it will result in using another algorithm to truncate them. Also, we mentioned earlier that due to high speed, good security, and high level of public acceptance, we will use the AES algorithm for encryption, and specifically we will use the AES-128 method for our proposed mechanism that has 16-byte encryption output. At the end, we have placed an 8-byte SSV in the middle of the payload part of the message frame, which is next to the message body that is encrypted, and a MAC is made from it. This MAC value is initially appended to the entire content in the payload. The point is that this MAC with a length of 8 bytes is the result of the SipHash algorithm. The length of the original message included at the end of the payload should also be equal to 16 bytes after encryption. We tend to use this value of 16 bytes in our proposed mechanism. But the main point in this method, compared to other methods that we reviewed in related works, is the use of the Encrypt Then MAC method. The purpose is to better prevent data tampering attacks. First, we encrypt the content of the original message that contained the main message with AES-128 and then use message digest with a SipHash MAC. With this method, on the receiver's side,

if a tampered message has been sent to it, it does not need to decrypt the message to test the integrity after that, and it can directly test the integrity at the moment. The final message frame is in the form of Fig 5 which is similar to Fig 1, but the secure payload is MAC || SSVu || E (Key [Original Message]).

Preamble 1-byte	Address 3-5-Byte	(PCF) 9-bits	Secure Payload	CRC 1-2-Bytes

Fig 5 Final proposed message frame for the Enhanced Shock Burst protocol.

Also, the proposed secure payload of our security mechanism is shown in Fig 6.

MAC Digest 8-Byte	SSVu 8-Byte	E (Key [Original Message]) 16-Byte

Fig 6 The proposed secure payload in separate parts.

We use a block diagram to depict our mechanism using the GPS module as stated in Fig 7 As can be seen, in the block diagram shown in Fig 7 we assumed that we have a transmitter device and a receiver device, both of which are based on the ESB network communication protocol and use the nRF24L01 communication module. We can use a separate transmitter and receiver or a transceiver. We should have two devices as transceivers to send or receive data simultaneously.

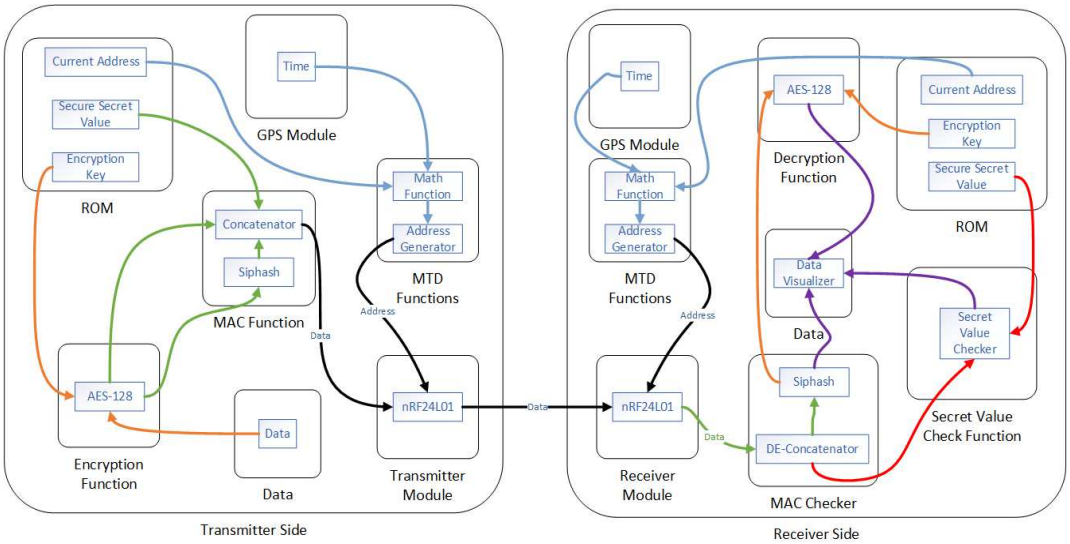

Fig 7 Block diagram of the proposed mechanism.

As we can see in the pseudo-code in Table 3., at first the encryption key and SSV value will be generated by a trusted authority and will be embedded in the microcontrollers of transmitters and receivers. On both the transmitter and receiver side, while turning on the modules, the current time is received from GPS and a new address is generated every three minutes. then, from the transmitter, the message M is encrypted and the MAC digest is generated from the cipher text (Encrypt then MAC). Both the encrypted message and the MAC and the SSV will be sent to the receiver. On the receiving side, the message is broken into three parts, first, the MAC value is confirmed, then the SSV value is confirmed and if both are confirmed, the original message is decrypted and will be used.

As shown in Table 4, due to the presence of the address shuffling technique and its use, it can be possible to avoid all kinds of DoS attacks to a large extent. In fact, despite such a feature, the adversary must look for the data transfer address in a 5-byte address space. If an attacker finds it, he will have a limited opportunity to analyze the network because the address is going to shuffle again soon.

Using the Mobius function, we tried to create a condition in the address shuffle behavior where the NA cannot be easily guessed and to create a kind of chaos. Therefore, the attacker has to go through a complex process with a limited opportunity to reach our CA. In this case, we can claim that we have created a suitable defense solution against DoS attacks. In the next step, if the attacker can guess the CA and even a sequence of addresses, we will have the capacity to fall prey to a DoS attack. However, we have considered some tests using some artificial intelligence systems to guess our address sequence in different situations, but they didn't succeed.

Table 3 Pseudo-code of the proposed security mechanism.

Registration Center as Trusted Authority	Transmitter Side	Receiver Side
+ Generate SSV_u, Encryption Key, and MAC Key. + Implement SSV_u and Keys on devices.		
	+ Get the time of GPS. + Each 3 minutes: compute PA, μ(PA), NA and shuffle address. While using the new address: + Generate new message M. + Compute M'= E (Key [M]) using AES-128. + Compute MAC(M') using Siphash algorithm. + M"= MAC (M') ‖ SSV_u ‖ M'. + Sending M" to Receiver using nRF24L01.	+ Get the time of GPS. + Each 3 minutes: compute PA, μ(PA), NA and shuffle address.
		While using the new address: + Get M" from the Transmitter using nRF24L01. + De-Concatenate Message into 3 parts that M"= MAC(M')/ SSV_u /M'. + Verify SSV_u. + Verify the MAC of M' using the Siphash algorithm. + M= Decrypt M' using AES-128. + Use command or data in M.

Success attacks do not happen as long as our variables, including the minute and initial address, remain confidential. This means that a person cannot guess that one of the influencing variables in shuffling the address is time, and there is also an initial address value. Apart from that, due to the same address shuffle mechanism, the occurrence of other attacks will also be a serious challenge for the attacker. Specifically, attacks that affect wireless communication modules based on ESB communication protocol can be faced with a serious challenge with this technique, except for attacks based on sending noise or jamming, which these solutions are ineffective to resist. On the other hand, if, assuming, the attacker can obtain details of our network by analyzing these conditions and forming the address sequence, using proper encryption based on AES-128, he will lose the ability to perform MITM attack or eavesdropping attack operations as long as our main key will be revealed. One of the features of AES-128 encryption is the guarantee of confidentiality, as long as the encryption key is not exposed. Also, due to the presence of the SipHash-type MAC digest, the possibility of attacks such as tampering is eliminated, and the attacker can hardly destroy the integrity of our message-sending process, as long as he does not know we use the SipHash algorithm and what is the key. Another technique in this method compared to other methods is using encryption and MAC. In the common approaches that you saw some of these in the related works part, first, we saw the MAC digest setting and

then the encryption process was applied to the combination of MAC and the original message. To establish security against identity spoofing or impersonation and Sybil Attacks, we use an SSV as device information to authenticate that device in the network. In the end, since the device's SSV is stored in the microcontrollers and also the presence of the authentication mechanism, the occurrence of Sybil attacks will also be a serious challenge for the adversary because, for each message sent in the network we have authentication on the agenda. It means a malicious node cannot connect several other malicious nodes to our system without having the right amount of SSV. In this mechanism, we manage to create a secure wireless channel while guaranteeing confidentiality, information integrity, time synchronization, accountability, and availability.

Table 4 Comparison of establishing a security mechanism for the proposed method with the previous methods.

Security Requirements & Features Against Various Attacks	[17]	[18]	[19]	[20]	Proposed Mechanism
Accountability Requirement (SR-1)	No	No	No	No	Yes
Integrity Requirement (SR-2)	Yes	No	Yes	No	Yes
Confidentiality Requirement (SR-3)	Yes	Yes	Yes	Yes	Yes
Availability Requirement (SR-4)	No	No	No	No	Yes
Moving Target Defense (SF-1)	No	No	No	No	Yes
User/Node Authentication (SF-2)	No	No	No	No	Yes
Eavesdropping Attack (SF-3)	Yes	Yes	Yes	Yes	Yes
Man in The Middle Attack (SF-4)	Yes	Yes	Yes	Yes	Yes
Sybil Attack (SF-5)	No	No	No	No	Yes
Impersonation / Identity Spoofing Attack (SF-6)	No	No	No	No	Yes
Denial of Service (DoS) Attack (SF-7)	No	No	No	No	Yes
Password Guessing Attack (SF-8)	No	No	No	No	Yes
Tampering Attack (SF-9)	Yes	No	Yes	No	Yes
SYN Flood Attack (SF-10)	No	No	No	No	Yes

6. Implementation and Experimental Results

To implement our proposed mechanism, we used the Arduino platform and specifically the Uno type consisting of the Atmega328p microcontroller. Among the reasons for using the Arduino platform are the reasonable price, sufficient processing power due to the lack of high expectations in the laboratory process, ease of using the platform (especially the number of libraries available for this platform), and efficiency in laboratory processes.

Nevertheless, it is recommended to industrialize the existing design, while testing the microcontroller that supports our mentioned method. Depending on the amount of memory usage, processing power, power consumption, and the number of analog and digital pins on the microcontroller, supported technologies and the proposed libraries will be selected by the development team. Also, to benefit from GPS protocol, we used the NEO6M GPS module, which acts as a third party to receive time. This module besides receiving geographic data also has unique capabilities such as determining height above sea level, latitude, longitude, speed of movement, and instantaneous Greenwich time in normal or raw form [43]. The NEO6M module uses the I2C communication protocol to connect to other electronic components, including microcontrollers. Modules that support this type of communication have two TXD and RXD pins next to two power supply pins VCC and GND. The information is sent from the TXD pin to the microcontroller, and through the RXD pin, it receives the information it needs from the microcontroller.

Also, to work with the ESB communication protocol, we used the nRF24L01 module, which is capable of transmitting information at different rates up to 2 Mbps. Also, if we need the quality or communication range to increase, we can use other modules of this family, such as nRF24L01+PA+LNA.

Using Fritzing software, we can draw the electronic design and map the connections of electronic devices to each other. In Table 4. and Fig 8 in detail and Fig 9 as a prototype implementation, how to connect various electronic components to Arduino can be seen for a transceiver node in the network.

Keep in mind that in this design, we have directly supplied the energy supply of the two communication modules Neo6M and nRF24L01 from the microcontroller, but in a more efficient mode, it is better not to supply the energy from the microcontroller. On the other hand, it is important to know that the working voltage of Neo6M is equivalent to 5 volts and the working voltage of nRF24L01 is equivalent to 3.3 volts.

It is worth noting that nRF24L01 communication modules benefit from distributed capability by default and you can finally benefit from many-to-many communication in them [44].

Table 4 Circuit connections to build a node with nrf24l01 communication module & neo6m GPS module with Arduino.

Port Type	Port Pin	Arduino Uno Pin
nRf24L01	MISO	12
	SCK	13
	CE	9
	MOSI	11
	CSN	8
	VCC	3v3
	GND	GND
NEO6M	TXD	4
	RXD	3
	VCC	5V
	GND	GND

To implement the proposed security mechanism on the Atmega328p microcontroller in Arduino, we used two libraries "SipHash-2-4" and "AESLib Master" in the Arduino IDE environment in C++ language. Iin practice, our proposed method showed a very good speed of operation. In our proposed method, the time needed for each encryption with the AES-128 algorithm is only 284 microseconds and the time to calculate the MAC digest is only 864 microseconds, which together takes 1148 microseconds, as we can see in Figs 10 and 11.

In terms of performance time spent compared to the paper [19], our proposed method provides a higher performance speed on a completely similar platform as shown in Fig 10.

Also, in terms of performance speed, our proposed method can be compared with the [17] with the assumption that we know that the microcontroller used in their research is almost two times weaker than the microcontroller used by us, but the comparison of both samples in terms of performance speed, while the method [17] uses 5 working modes and we only use one mode, is significant as it is shown in Fig 11.

In the end, it should be mentioned that in the circuit we built, regardless of the address shuffle process and the presence of the GPS module, the average current consumption is equal to 0.33 mA and if we add the GPS module for frequent address shuffles, it will have an average current consumption of 0.90 mA.

Also, the power consumption is 3.33 mW with GPS and address shuffling and 1.11 mW without GPS and address shuffling.

Also, in Fig 12 we have the cost comparison of our proposed systems with some other methods.

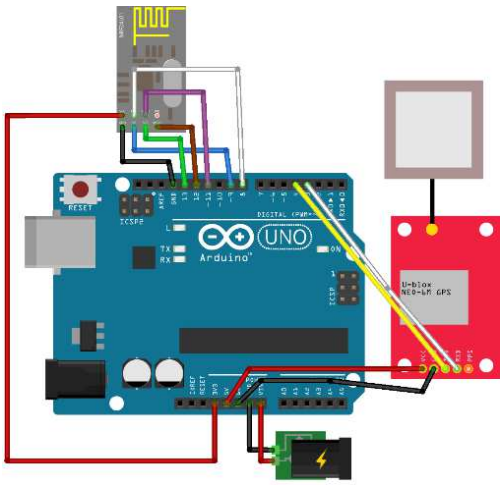

Fig 8 The proposed prototype circuit.

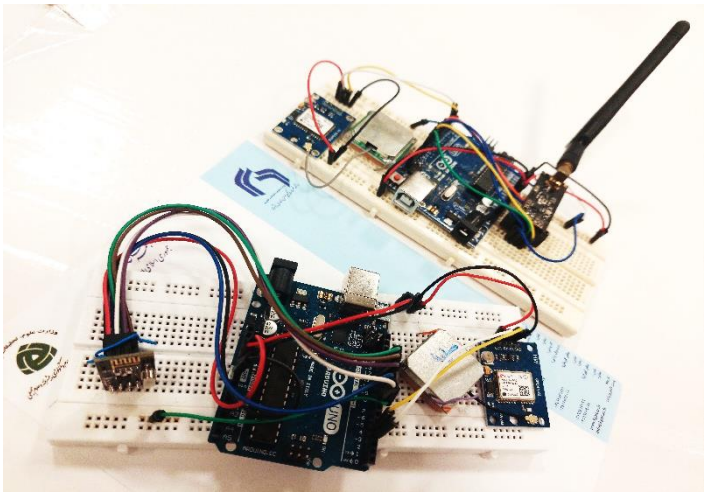

Fig 9 Implemented Prototypes.

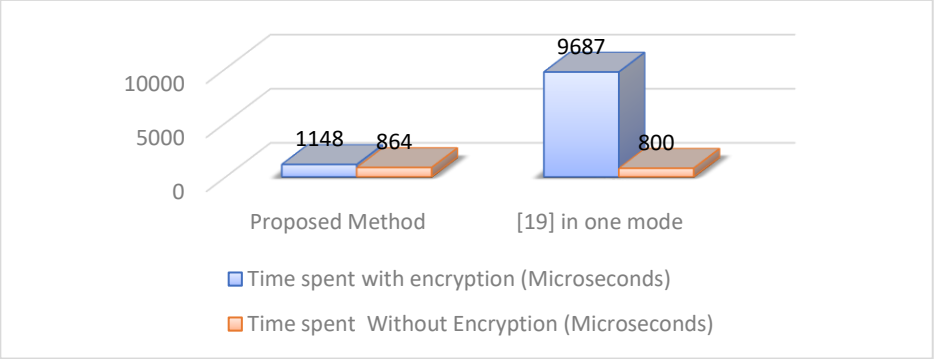

Fig 10 Time spent of proposed method in comparison with [19].

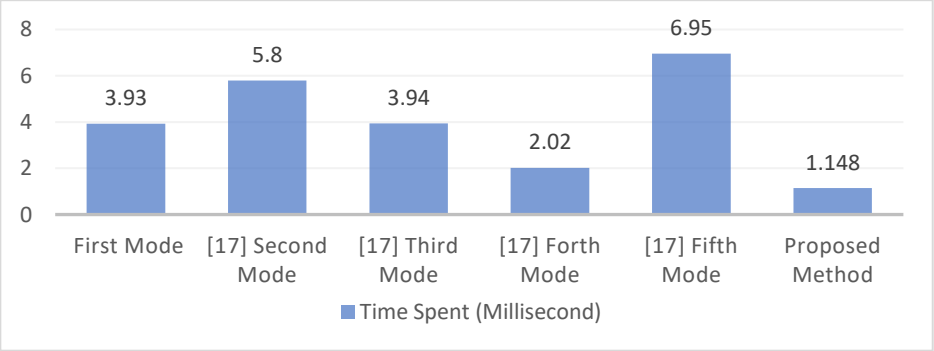

Fig 11 Time spent of proposed method in comparison with [17].

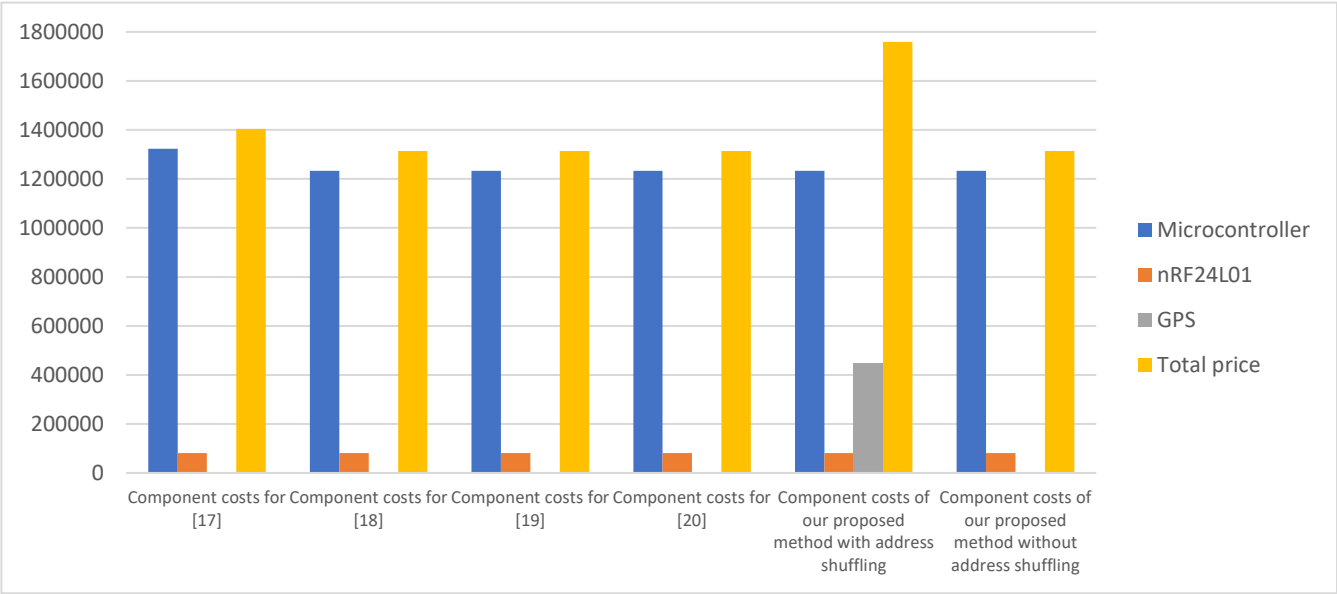

Fig 12 Components cost comparison diagram

7. Conclusion and Future Works

The importance of establishing security in the fields of the IoT, WSNs, and all related fields is not a secret to anyone these days. In this article, efforts were made to provide confidentiality, authentication, message integrity, and a high level of accessibility along with appropriate speed and energy consumption. Previously, many security schemes were presented for different network protocols in the field of IoT and WSNs, but the security issues of our proposed method have some differences. In our proposed Mechanism, which is based on the ESB communication protocol, we were able to provide a security mechanism that, in addition to being lightweight, is capable of being used in any application of the ESB communication protocol, including drones and especially unmanned aerial vehicles (UAVs), mechanical systems, industrial systems, IoT, WSN play a role in a safer way than before.

For future work, we believe that a mechanism to deal with Jamming attacks should be thought of. In addition, if very fast and secure accepted encryption methods are provided, they can be used instead of AES-128. On the other hand, if a better algorithm is created than Siphash, which allows higher speed, more security, and a suitable length of the MAC digest, it can be replaced by Siphash. Also in authentication, to increase the security of the transmitted SSV on the network, we can use SALT. Also, to deal with replay attacks, the time stamp can be added to the packet to prevent replay attacks.

With all interpretations, our proposed mechanism for ESB communication protocol responded well and we managed to create a secure channel that seems to be safe against the mentioned attack example, in one method, provided that we know what the initial address is and also we know the next two or three addresses, not only we could not form the correct address sequence based on our mathematical knowledge with mathematical functions, but also with Bing-based artificial intelligence systems also could not form a range of addresses correctly and failed. In the end, it is worth mentioning that about the address shuffle method, for long-term communications, a time threshold should be defined so that the addresses are reset and the output of the address shuffling function does not include very large numbers.

References

1. Khan, M. A., & Abuhasel, K. A. (2021). Advanced metameric dimension framework for heterogeneous industrial Internet of things. *Computational Intelligence, 37*(3), 1367-1387.
2. Ojha, T., Misra, S., & Raghuwanshi, N. S. (2021). Internet of things for agricultural applications: The state of the art. *IEEE internet of things Journal, 8*(14), 10973-10997.
3. Khan, M. A. (2020). An IoT framework for heart disease prediction based on MDCNN classifier. *Ieee Access, 8*, 34717-34727.
4. Estrada-López, J. J., Castillo-Atoche, A. A., Vázquez-Castillo, J., & Sánchez-Sinencio, E. (2018). Smart soil parameters estimation system using an autonomous wireless sensor network with dynamic power management strategy. *IEEE Sensors Journal, 18*(21), 8913-8923.
5. Hassan, W. H. (2019). Current research on Internet of Things (IoT) security: A survey. *Computer networks, 148*, 283-294.
6. Bálint, Á., & Sárosi, J. (2016). Design and Implementation of a radio controlled led lighting system. *Analecta Technica Szegedinensia, 10*(1), 29-34.
7. Ghosh, S., Ghosh, K., Karamakar, S., Prasad, S., Debabhuti, N., Sharma, P., et al. Development of an IOT based robust architecture for environmental monitoring using UAV. In *2019 IEEE 16th India Council International Conference (INDICON), 2019* (pp. 1-4): IEEE
8. Lv, D., Liang, C., & Zhang, Y. (2023). Research and Design of Four Rotor UAV Based on Cascade PID. In *Advances in Artificial Systems for Medicine and Education VI* (pp. 88-99): Springer.
9. Kulasekara, V., Balasooriya, S., Chandran, J., & Kavalchuk, I. Novel low-power NRF24L01 based wireless network design for autonomous robots. In *2019 25th Asia-Pacific Conference on Communications (APCC), 2019* (pp. 342-346): IEEE
10. Luo, Y.-B., Wang, B.-S., Wang, X.-F., Hu, X.-F., Cai, G.-L., & Sun, H. RPAH: Random port and address hopping for thwarting internal and external adversaries. In *2015 IEEE Trustcom/BigDataSE/ISPA, 2015* (Vol. 1, pp. 263-270): IEEE
11. MacFarland, D. C., & Shue, C. A. The SDN shuffle: Creating a moving-target defense using host-based software-defined networking. In *Proceedings of the Second ACM Workshop on Moving Target Defense, 2015* (pp. 37-41)
12. Wang, F., Wang, H., Wang, X., & Su, J. (2012). A new multistage approach to detect subtle DDoS attacks. *Mathematical and Computer Modelling, 55*(1-2), 198-213.
13. Benyezza, H., Bouhedda, M., Kara, R., & Rebouh, S. (2023). Smart platform based on IoT and WSN for monitoring and control of a greenhouse in the context of precision agriculture. *Internet of Things, 23*, 100830.
14. Mahbub, M. (2020). A smart farming concept based on smart embedded electronics, internet of things and wireless sensor network. *Internet of Things, 9*, 100161.
15. Zhang, L., Suzuki, H., & Koyama, A. (2021). Recognition of meal information using recurrent neural network and gated recurrent unit. *Internet of Things, 13*, 100358.
16. Sokullu, R., Akkaş, M. A., & Demir, E. (2020). IoT supported smart home for the elderly. *Internet of Things, 11*, 100239.
17. Selimis, G., Huang, L., Massé, F., Tsekoura, I., Ashouei, M., Catthoor, F., et al. (2011). A lightweight security scheme for wireless body area networks: design, energy evaluation and proposed microprocessor design. *Journal of medical systems, 35*(5), 1289-1298.
18. Babu, P. S., & Panda, B. S. Light Weight Security and Authentication in Wireless Body Area Network (Wban). In *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), 2020* (pp. 1-7): IEEE
19. Rivera, D., García, A., Martín-Ruiz, M. L., Alarcos, B., Velasco, J. R., & Oliva, A. G. (2019). Secure communications and protected data for a Internet of Things smart toy platform. *IEEE internet of things Journal, 6*(2), 3785-3795.
20. Kanthi, M., & Dilli, R. (2023). Smart streetlight system using mobile applications: secured fault detection and diagnosis with optimal powers. *Wireless Networks*, 1-14.
21. Team, M. E. (2015). Ending support for the RC4 cipher in Microsoft Edge and Internet Explorer 11. https://blogs.windows.com/msedgedev/2015/09/01/ending-support-for-the-rc4-cipher-in-microsoft-edge-and-internet-explorer-11/.
22. Liu, Y., & Han, X. Analysis of the maximal transmission rate based on nRF24L01 chip system. In *2010 2nd International Conference on Information Engineering and Computer Science, 2010* (pp. 1-3): IEEE
23. In-Depth: How nRF24L01+ Wireless Module Works & Interface with Arduino (2018). https://lastminuteengineers.com/nrf24l01-arduino-wireless-communication/.
24. nRF24L01. https://www.nordicsemi.com/products/nrf24-series.
25. Walters, J. P., Liang, Z., Shi, W., & Chaudhary, V. (2007). Wireless sensor network security: A survey. In *Security in distributed, grid, mobile, and pervasive computing* (pp. 367-409): Auerbach Publications.
26. William, S. (2011). *Network Security Essentials: Applications and Standards (For VTU)*: Pearson Education India.
27. Van Steen, M., & Tanenbaum, A. S. (2017). *Distributed systems*: Maarten van Steen Leiden, The Netherlands.
28. Tsao, K.-Y., Girdler, T., & Vassilakis, V. G. (2022). A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks. *Ad Hoc Networks, 133*, 102894.
29. Wheeler, E. (2011). *Security risk management: Building an information security risk management program from the Ground Up*: Elsevier.
30. van Oorschot, P. C. (2020). *Computer Security and the Internet*: Springer.
31. Halak, B., Yilmaz, Y., & Shiu, D. (2022). Comparative analysis of energy costs of asymmetric vs symmetric encryption-based security applications. *Ieee Access, 10*, 76707-76719.

32. Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. Comprehensive study of symmetric key and asymmetric key encryption algorithms. In *2017 international conference on engineering and technology (ICET), 2017* (pp. 1-7): IEEE
33. Mousavi, S. K., Ghaffari, A., Besharat, S., & Afshari, H. (2021). Security of internet of things based on cryptographic algorithms: a survey. *Wireless Networks, 27*, 1515-1555.
34. Abood, O. G., & Guirguis, S. K. (2018). A survey on cryptography algorithms. *International Journal of Scientific and Research Publications, 8*(7), 495-516.
35. NIST (2022). NIST Retires SHA-1 Cryptographic Algorithm. https://www.nist.gov/news-events/news/2022/12/nist-retires-sha-1-cryptographic-algorithm.
36. Hagenlocher, P. (2018). Performance of message authentication codes for secure ethernet. *Network, 27*.
37. Aumasson, J.-P., & Bernstein, D. J. SipHash: a fast short-input PRF. In *International Conference on Cryptology in India, 2012* (pp. 489-508): Springer
38. Wang, T.-Z., Wang, H.-M., Liu, B., Ding, B., Zhang, J., & Shi, P.-C. (2012). Further analyzing the sybil attack in mitigating peer-to-peer botnets. *KSII Transactions on Internet and Information Systems (TIIS), 6*(10), 2731-2749.
39. Zheng, Y., Li, Z., Xu, X., & Zhao, Q. (2022). Dynamic defenses in cyber security: Techniques, methods and challenges. *Digital Communications and Networks, 8*(4), 422-435.
40. Cai, G., Wang, B., Wang, X., Yuan, Y., & Li, S. An introduction to network address shuffling. In *2016 18th international conference on advanced communication technology (ICACT), 2016* (pp. 185-190): IEEE
41. He, D., & Hu, H. (2013). Cryptanalysis of a dynamic ID-based remote user authentication scheme with access control for multi-server environments. *IEICE TRANSACTIONS on Information and Systems, 96*(1), 138-140.
42. He, D., Wu, S., & Chen, J. (2012). Note on 'Design of improved password authentication and update scheme based on elliptic curve cryptography'. *Mathematical and Computer Modelling, 3*(55), 1661-1664.
43. Buchli, B., Sutton, F., & Beutel, J. GPS-equipped wireless sensor network node for high-accuracy positioning applications. In *Wireless Sensor Networks: 9th European Conference, EWSN 2012, Trento, Italy, February 15-17, 2012. Proceedings 9, 2012* (pp. 179-195): Springer
44. Gia, T. N., Dhaou, I. B., Ali, M., Rahmani, A. M., Westerlund, T., Liljeberg, P., et al. (2019). Energy efficient fog-assisted IoT system for monitoring diabetic patients with cardiovascular disease. *Future Generation Computer Systems, 93*, 198-211.

Aref Ayati graduated from Isfahan University with a B.Sc in computer sciences in 2020, and he earned an M.Sc degree in Information Technology Engineering in 2023 from Iran's Graduate University of Advanced Technology, Kerman. His areas of interest include IT management, IoT, WSNs, UAV Systems, and Computer Networks and security.

Hamid Reza Naji is associate professor of computer engineering at the Graduate University of Advanced Technology, Iran. His research interests include embedded systems, distributed, parallel and multi-agent systems, networks, and security. Dr. Naji has a Ph.D. in computer engineering from the University of Alabama in Huntsville, USA.