Journal of Information Systems and Informatics
Vol. 6, No. 4, December 2024 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v6i4.886 Published By DRPM-UBD

Balancing Energy Efficiency and Security in CR-LoRaWAN Ecosystems

Koketso Ntshabele[1], Bassey Isong[2,*]

[1,2] Department of Computer Science, North-West University, Mafikeng, 2790, South Africa
Email: [1]koketso.ntshabele@nwu.ac.za, [2,*]bassey.isong@nwu.ac.za

Abstract

Cognitive Radio-enabled Long Range Wide Area Networks (CR-LoRaWAN) plays an important role in IoT applications. However, due to the limitations of devices and dynamic scheduling mechanisms of the channels, there is still a challenge to balance energy efficiency against security. This paper proposes two developed algorithms that address these challenges: Algo A and Algo B. Algo A ensures key security by mitigating nonce generation vulnerabilities through the replacement of insecure random numbers with prime numbers. Algo B develops this basis by further improving energy efficiency through optimization in session key generation and device management, adding security to it. Both the algorithms incorporate prime numbers in their session key generation that are verified by the Rabin-Miller test and the Sieve of Eratosthenes, with incorporated solar energy harvesting to give a longer life to such devices. Cognitive radio technology is integrated into it for dynamic and intelligent channel selection. Extensive simulations demonstrate that Algo A is much better at handling data with key security, while Algo B outperforms Algo A on energy consumption reduction by 20% and enhancement of overall network security by 15%. These results reveal that Algo B has a better trade-off between security and energy efficiency; hence, Algo B is more suitable for practical deployment. The work further enhances the sustainability and reliability of CR-LoRaWAN networks, especially in resource-constrained environments.

Keywords: Cognitive Radio, IoT, LoRaWAN, Prime numbers, Energy Efficiency, Energy harvesting, Key Security, Algo A, Algo B

1. INTRODUCTION

1.1 Introduction and Background

Internet of Things-IoT can be defined as an assembly of several devices comprising sensors, software, and various other technologies which collect data and send it over the Internet. Devices will span a wide range of applications, from smart homes, smart cities, and manufacturing to agriculture, healthcare, and transportation [1-4]. Although IoT can be a game-changer for these industries, it has its challenges, such as resource-constrained devices and potential security issues. In meeting these challenges, LoRaWAN was developed. LoRaWAN is a Low Power Wide Area Network protocol developed explicitly for IoT applications. It ensures large coverage, long-range communication with energy efficiency, and cost-effectiveness [3, 4]. LoRaWAN, being based on the long-range protocol, offers license-free sub-gigahertz radio frequency bands. It thus allows interoperability between network components from different vendors and, with open-source technology, makes implementation of IoT applications easier. LoRaWAN can support more than 10 kilometres under favourable conditions, while data rates vary from 0.3 kbps to 50 kbps per channel due to the spreading factor. With that in consideration, LoRaWAN is particularly fit for almost any type of IoT application in remote areas where cellular is poor or places where hardly any other communication can take place. However, LoRaWAN also suffers from some existential challenges in particular energy efficiency and security, which become highly paramount for resource-constrained end devices whose communication contains sensitive data. In as much as LoRaWAN has integrated those mandatory authentication and encryption mechanisms, there still exists a probability of security vulnerability if the cryptographic nonces get reused or if keys have not been properly distributed across devices. Furthermore, key management issues, flawed implementations, and vulnerabilities to attacks are significant concerns [7, 8, 10]. While AES-128 symmetric encryption is the standard for securing LoRaWAN communications, the efficiency of key management remains a challenge, as no universal security model has emerged to address all potential threats [4, 5, 7-9].

Moreover, the integration of Cognitive Radio (CR) technology into LoRaWAN offers the potential for improved spectrum efficiency by enabling dynamic spectrum access, allowing devices to adaptively select less congested communication channels. However, the adoption of CR introduces new challenges in real-time spectrum sensing, adaptive channel selection, and securing communications in a rapidly changing spectrum environment. This dynamic nature complicates the implementation of traditional security protocols, which may not be agile enough to protect communications effectively. With the evolution of CR-LoRaWAN networks, securing those networks in an energy-efficient manner will be more challenging because the underlying IoT devices have limited processing capability. Algo_A and Algo_B try to fill this gap by proposing optimized security mechanisms combined with the dynamic nature of CR while guaranteeing energy efficiency. This work, therefore, develops two algorithms, namely Algo_A and Algo_B. Both would advance energy efficiency and improve the key security in CR-LoRaWAN. First, Algo_A will try to enhance key management and distribution through a trusted key server in a centralized approach for efficient and secure encryption key handling. Algorithm Algo_B improves the basis set by Algo_A through the patching up of important weaknesses and the addition of advanced features, including more secure key secrecy and randomizing channel allocation. These algorithms use cognitive radio capability and energy-harvesting techniques to reduce power consumption considerably with no compromise in security; hence, they are suitable for practical IoT applications over CR-LoRaWAN networks. This research therefore contributes, the significance of which is central to the whole IoT ecosystem. As CR-LoRaWAN networks evolve further, the ability to achieve energy efficiency along with robust security will be major milestones for widespread adoption across a range of applications. Energy-efficient and secure networks open new opportunities for deploying IoT solutions across remote rural to urban environments characterized by reliable communication, long device lifespans, and low maintenance costs. The improvements of Algo_B, in terms of enhanced throughput, security, and processing speed, reflect a radical improvement compared to existing current models. These will facilitate the practical functioning of smart cities, environmental monitoring systems, precision agriculture, and other IoT-driven applications even in resource-constrained environments. For this, both Algo_A and Algo_B have ensured considerable contributions toward the optimization of the energy and security mechanism within a CR-LoRaWAN network. These algorithms facilitate a better management capability of large-scale IoT deployments while keeping both operational efficiency and data integrity at bay.

1.2 LoRaWAN and Security

LoRaWAN is a wireless communication paradigm designed for IoT devices requiring long-range connectivity with low power consumption. Operating on unlicensed radio channels allows LoRaWAN to deploy low-cost IoT networks across large geographical areas. LoRaWAN uses a star-of-stars topology where gateways forward data from transmitting entities to a centralized network server. It ensures the integrity, availability, and security of transmitted data. LoRaWAN is designed to reduce these threats using end-to-end encryption. The encrypted data from the transmitting end device is decrypted only by the authenticated recipient. Each device uses different sets of application and session keys to mutually authenticate each other for secure communications. In this way, the security mechanism can ensure that any data exchanged by the gateways with the end devices is safe from unauthorized access and alterations. However, despite these measures for the security of LoRaWAN, it is not free from vulnerabilities either [8], [10]. Like other IoT systems, it is susceptible to replay attacks, man-in-the-middle attacks, and device tampering [7,8,10].

1.3 Security Models

To achieve optimal network security and effectiveness, the following primary security models in CR-LoRaWAN are examined:

1) Rabin Miller Primality Test

Prime numbers are crucial for generating security keys, making them essential in cryptography [11,12]. Cryptographic systems require extremely large prime numbers for encryption, focusing on decomposing massive numbers into prime factors. Prime determinants are algorithms that identify numbers divisible only by one and themselves. "K-rounds" refer to the rounds used in stochastic primality tests.

2) Hash Functions

Hash functions convert input data into a fixed-size string of characters called a hash code. Each input generates a unique output. For example, SHA-256 algorithms are used in critical security applications as they produce outputs that are difficult to reverse-engineer, ensuring data integrity by preventing bit flipping [9]. Hash functions are widely used in digital signatures, password security, and secure communications to ensure data consistency [9]. Hash-based Message Authentication Code-based Key Derivation Function (HKDF): HKDF uses two phases: extraction and expansion. In the extraction phase, a pseudorandom key is generated from the master key while the pseudorandom key is used to create multiple keys for different applications in the expansion phase [9]. This ensures communication integrity and confidentiality which keep newly generated keys secure and unique [9].

3) Cognitive Radio

CR was developed in wireless technology to solve underutilized spectrum and interference problems that are the usual scenarios affecting radio communication systems. It is smart because it works in an automated manner, switching itself to open spectrum bands whenever necessary, thus adaptable in real-time to happenings within the environment. This flexibility is highly significant as more and more wireless devices and applications are now coupled, for which the traditional approaches of spectrum management are becoming less effective, hence resulting in congested and inefficient channels [13-16]. The technology behind CR is powered by software-defined radio, which allows the flexible and adaptive operation of radio systems [13-15]. Thus, CR devices can scan their surroundings independently, find less busy channels, and switch to them to avoid interference, therefore helping to keep network performance at a good level when there is heavy traffic in the network area. Due to the more efficient usage of the available spectrum, CR contributes to the creation of more reliable and robust networks, enabling further development of new wireless applications such as IoT and future wireless networks. This work proposes two enhanced algorithms, namely Algo_A and Algo_B, for better key management in Cognitive Radio-based LoRaWAN. These solutions will draw a balance between energy efficiency and security of keys over LoRaWAN networks, as discussed in [9]. This balance is achieved using CR for dynamic channel selection, the employment of solar energy harvesting to extend the battery life of end devices, and the implementation of hash functions and maps for efficient device and address management. More importantly, a central trusted key server is used to generate and manage encryption keys within the CR-LoRaWAN model that would be propagated to all connected devices [9]. On the other hand, Algo_B is the enhanced version of Algo_A. It has been specifically designed to mitigate the security vulnerabilities of Algo_A and, accordingly, introduce a novel secured model [9]. In this respect, it uses CR capabilities for adaptive and randomized channel allocation to support the inherent principles of efficiency and fairness correspondingly. Additionally, it leverages solar energy harvesting and hashing techniques, which have been applied for the efficient addressing of devices and management of data [9]. For the effectiveness evaluation of these algorithms, a series of simulations was performed in the environment introduced by JAVA using the integrated environment called Eclipse IDE. Moreover, the Scyther tool was used to verify key secrecy against both unknown and known key attacks, inside and outside the expected parameters. From the analysis, the results obtained would indicate that Algo_B is performing better compared to Algo_A in terms of throughput as well as in terms of processing speed. This makes it a very good candidate for real-world applications requiring performance and strong security. The model provides new significant improvements to the existing knowledge in this domain.

The remainder of the paper is organized as follows: Section 2 discusses the methodology and related works, Section 3 presents the proposed secure and lightweight energy-efficient CR-LoRaWAN, Section 4 presents simulations and results, and Sections 5 and 6 provide the paper discussion and conclusion, respectively.

2 METHODS

The integration of LoRaWAN and CR technology to create CR-LoRaWAN networks introduces novel approaches to advance IoT efficiency and security. This section discusses the related works, simulation environment and parameters used to analyze and evaluate our proposed Algo_A and Algo_B-based CR-LoRaWAN networks, and last, is the design and development of the proposed solution as illustrated in Figure 1. The objective lies in addressing key distribution, energy efficiency, and channel management under high interference and fluctuating conditions by leveraging CR capabilities for session key transmission. Thus, the implementation of Algo_A and Algo_B was conducted using a simulation tool, not a testbed application, to model and analyze the behaviour of the proposed CR-LoRaWAN network. This study is aimed at understanding its impact on various parameters before validating its applicability in real-world applications, as we have recommended for future research.

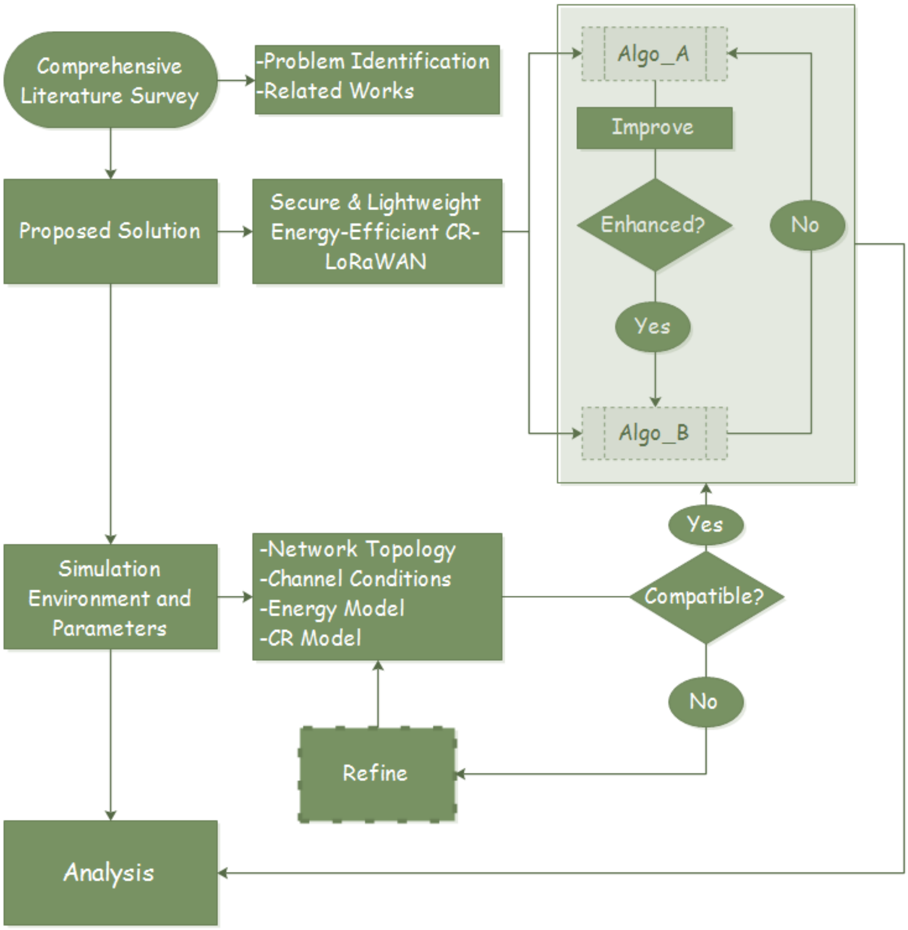

Figure. 1. Methods

2.1 Related Works

This subsection presents a discussion of existing studies, and the challenges faced by these networks. Ntshabele et al. [9] addressed security challenges in LoRaWAN using AES 128-bit symmetric keys and proposed two new initial security algorithms, Algo_A and Algo_B, to improve key management. However, their study lacks full verification and does not balance security and energy efficiency, which is the critical focus of this paper to provision a more thorough approach based on verification. Similarly, Salika et al. [13] suggested integrating CR and LPWAN to simplify resource-constrained sensor nodes, improving both the physical layer and network architecture. They addressed channel congestion and interference using adaptive threshold techniques but did not fully explore the security and energy efficiency trade-off, a gap our paper aims to fill.

In a similar study, Muteba et al. [14] suggested a Narrowband CR-IoT model using a Deep Q-learning algorithm to minimize repeated transmissions and spectrum wastage. While their approach is more secure than traditional Q-learning algorithms, it does not address specific security challenges related to key management, which Algo_B in our paper specifically targets. Again, a channel-sensing model based on energy detection in CR-LPWAN was developed by Mruoe et al. [15], to improve outdoor coverage in urban areas and prolong end devices' lifespan using an Eligibility Score approach. On the same note, Umeonwuka et al. [16] presented dynamic spectrum access to optimize spectrum utilization in CR-LPWAN and improve network quality of service (QoS) and spectrum efficiency. However, key security management remains unexplored, which our paper addresses by introducing novel algorithms for secure key management and spectrum optimization. Kim and Song [4] also suggested a dual key-based activation system for automatic key updates in LoRaWAN, improving security through dynamic session key creation but not addressing energy efficiency. Our paper incorporates a lightweight key security model alongside solar energy harvesting for a holistic solution.

Furthermore, Tsai et al. [18] presented a secure low-power communication system based on AES-128 to balance security and power consumption in IoT networks, minimizing computational complexity and energy consumption. Similarly, Han and Wang [2] suggested a method for updating the LoRaWAN root key using a Rabbit Stream Cipher-Based Key Derivation Function scheme, enhancing security by reducing key derivation time and increasing randomness. These papers improve security but do not address the energy demands of IoT devices, an area our paper significantly improves with novel algorithms. Ruotsalainen et al. [19] equally suggested an LPWAN key manifestation technique to address channel probing and high latency in LoRaWAN class A devices, ensuring reliable key generation and transmission while minimizing packet collisions. Similarly, Roselin et al. [7] proposed a lightweight authentication protocol (LAUP) for 6LoWPAN Wireless Sensor Networks (WSN) to improve security without the use of pre-shared keys. The LAUP employs four flights for authentication and session key sharing, resulting in secure communication.

In the same vein, Naoui et al. [6] suggested a security model that has three phases: system setup, join procedure, and rekeying, where a trusted third party generates two session keys using asymmetric encryption to ensure secure communication between the third party and the network servers. The models proposed by [6] and [7] use a third party to enforce secure communication while often involving energy consumption and security trade-offs. In contrast, Algo_B in our paper minimizes these trade-offs by integrating an energy-harvesting model and a lightweight robust key secrecy model. Tsai et al. [8] also introduced a trusted third party-based high-efficiency multi-key exchange protocol to optimize user and message authentication in communication networks, thereby improving reliability, efficiency, and high throughput in key exchange processes. To mitigate these issues, Gao et al. [20] suggested a secure packet transmission (SPT) scheme to strengthen LoRaWAN 1.1 security during join request processes. The SPT uses redefined join request packets, one-time password encryption, and an adaptive data rate algorithm to enhance security and prolong the battery life of end devices. However, LPWAN solutions operating in ISM bands face issues such as spectrum interference, which can negatively impact the primary user's experience and QoS. Dönmez and Nigussie [23] analyzed attacks and countermeasures in LoRaWAN 1.1, with a particular emphasis on the join procedure. Their cryptographic process and key security measures contribute to strengthening network security against various attacks.

The above-highlighted studies comprehensively discussed the existing challenges and advancements in secure and energy-efficient IoT communication protocol devices [15, 16], [21, 22]. These studies aim to enhance energy efficiency and preserve the security and performance of the network. However, battery-powered IoT devices still face rapid energy depletion and spectrum interference challenges [16]. Dynamic key activation mechanisms and efficient encryption algorithms have been proposed [2,4,8] to enhance security while reducing energy consumption and computational complexity. Despite these advancements, efficient key management and secure communication are yet to be fully achieved in IoT networks [6,23]. The proposed CR-LoRaWAN solution in this paper aims to address the challenge of integrating energy-efficient models with intricate security protocols, thereby improving IoT network dynamism by balancing energy efficiency and key security.

2.2 Proposed Secure and Lightweight Energy-Efficient CR-LoRaWAN Model

In this section, we present a CR-LoRaWAN security model that relies on Algo_A and Algo_B. Our model uses the TKMS to handle session keys throughout the network. It is characterized by lightweight, security and energy efficiency, as well as employing the SHA-256 hash function to manage device identities and addresses. To maintain message integrity and authenticity, the model incorporates timestamps as well as utilises a solar energy harvesting system to extend the battery life of LoRaWAN devices. The integrated CR technology improves channel selection and scheduling for transmitting session keys, while the centralized key server generates and manages these keys to ensure secure and reliable communication.

Table 1. Theoretical State-of-the-Art Comparative Analysis

PROPOSED ALGO_A KEY SECRECY

```
Input: RequestToJoinMessage (EncryptedMessage, HashedMessage)
Output: AcceptToJoinMessage (HashedMessage)
While (RequestToJoin Initiated) {
  If (EncryptedMessage | HashedMessage) is transmitted between entities {
    //Key Generation
    Step1: EDs ⇨ KeySvr { (EncryptedMessage | HashedMessage) }
    Step2: KeySvr ⇨ NwkS { (EncryptedMessage | HashedMessage) }c
    Step3: NwkS ⇨ EDs { (EncryptedMessage | HashedMessage) }   }
  If (HashedMessage shared between sender and receiver) {
    // Performing Key Agreement Phase
    Step4&5: EDs ⇔ KeySvr { HashedMessage }
    Step6&7: EDs ⇔ NwkS { HashedMessage }
    Step8: EDs ⇨ KeySvr { HashedMessage }
    Step9&10: KeySvr ⇔ NwkS { HashedMessage }
    Step11: KeySvr ⇨ EDs { HashedMessage }   }
  If (EncryptedMessage | HashedMessage | HashedMessage approved by the
receiver until the      Step11) {
    // Authentication and Message Integrity Check Approved
    RequestToJoin Accepted
  } else {
// Terminate RequestToJoin: Unauthorized access and Falsified Message }}
```

1) Key security model

In the proposed CR-LoRaWAN-based security model, the key management protocol for Algo_A is presented in Table 1. The protocol works by guaranteeing communications between the KeySvr, EDs, and NwkS by encrypting and hashing transmitted messages. The process starts with the EDs sending an encrypted and hashed RequestToJoin message to the KeySvr to establish communication. The RequestToJoin message is considered valid and authentic only when the receiving entity verifies, confirms, and approves it. If the message is falsified or unauthorized, the transmission is terminated. In this case, the protocol mitigates security risks and guarantees data integrity by safeguarding the secure and authenticated joining of devices to the CR-LPWAN.

To address the persistent security challenges in the network, we proposed Algo_B by modifying and optimizing Algo_A. These modifications, depicted in Table 2, resulted in the development of Algo_B, which aims to overcome the identified security issues.

Table. 2. Proposed Algo_B Key Secrecy

```
PROPOSED IMPROVED ALGO_B KEY SECRECY
        Step3: NwkS ⇨ EDs { (EncryptedMessage | HashedMessage) }   }
    If (HashedMessage shared between sender and receiver) {
        // Performing Key Agreement Phase
        Step4&5: EDs ⇔ KeySvr { HashedMessage }
        Step6&7: EDs ⇔ NwkS { HashedMessage }
        Step8: EDs ⇨ KeySvr { HashedMessage }
       Step9&10: KeySvr ⇔ NwkS { HashedMessage }
```

To strengthen the security of Algo_A, Algo_B was refined from the 3rd to the 9th step. The modifications are as follows:

a) 3rd Step: The hash message transmitted from NwkS to ED is modified by replacing the EDs and NwkPrN parameters with EDsPrN. This change thwarts attacks such as bit flipping, ensuring the message's authenticity remains intact.

b) 6th Step: NwkPrN is replaced with EDsT to ensure that the acknowledgement message generated by NwkS is freshly created, thereby enhancing security for subsequent communications.

c) 7th Step: NwkST is replaced with NwkPrN to strengthen NwkS authentication using prime numbers.

d) 9th Step: An NwkID is integrated into the hash function to confirm that both NwkS and KeySvr share the same NwkSKey in the network. This improves the security of the newly generated message using KeySvrT.

2) The hash function and hash maps

In the proposed CR-LoRaWAN model, hash functions are critical to ensuring data integrity and managing keys. In this case, we use a Hash-based Message Authentication Code (HMAC) to establish cryptographic hashes. In this case, the hash function is combined with a secret key to maintain data integrity, security, and effective key management. In the same vein, we use hash maps to store hashed values and session keys, which simplifies the process of searching and verifying messages, helps prevent replay attacks, and supports efficient key management. For each received message, we initialize a hash map named message_i and iterate from 0 to 9 to cover all 10 different messages. Thus, each message is hashed and stored in the hash map, aiding in quick data retrieval.

The process begins by creating the HMAC keys required for the HMAC-based Key Derivation Function (HKDF). We generate these keys using the generateHMACKey() function, resulting in a unique secret key, hmacKey_i, for each transmission. This key is utilized to generate a secure cryptographic hash for further processing. We then extend the message with a hexadecimal string, hexEDsPrN, to produce concatenatedString_i. The SHA256 algorithm hashes this concatenated string to create hashedString_i. We then use the earlier generated

HMAC key in the HKDF to bind it with the byte representations of the sending and receiving agents, generating an AES key. This AES key is stored in the hash map message_i under the AES Key. Finally, message_i is created by concatenating concatenatedString_i with the derived AES key in its hexadecimal form. To ensure the message is properly verified and authenticated before sending, we include values in the message that specify the size of the AES key, the size of the hashed value, the total message size, and the total packet size. These values verify the integrity and structure of the message, ensuring all components are correctly sized and accounted for. This process secures and ensures reliable message transmission over the CR-LoRaWAN network. The hashing process is presented in Table 3.

Table. 3. Hashing Algorithm

```
PROPOSED CR-LORAWAN HASHING
1) Input 1: //Initialize HashMaps for messages
            HashMap<String , Object> message_i = new HashMap< >( );
                for i from 0 to 9:// i indicates the message number for each
      transmission
2) Input 2: //Initialize HMAC keys for use in HKDF
   a.  SecretKey = hmackKey_# = generateHMACKey( ); )//# indicates key number for each
       transmission
3) Processing: //Generate Message_i
   a.  concatenatedString_i = hexEDsPrN.toString() + Message_i;
   a.  hashedString_i = hashWithSHA256(concatenatedString_i);
   b.  derivedKey_i = deriveAESKey_Send_i(hmacKey_i, sending_entity_Bytes, receiving_entity_Bytes);
   c.  message_i.put("AES Key", derivedKey_i);
   d.  message_i.put("Hashed Value", hashedString_i);
   e.  Message_i = concatenatedString_i + bytesToHex(derivedKey_i);
   f.  Message_i = Message_i + hashedString_i;
4) Output: //Check and authenticate Message_i for sending
   a.  Print "Generated Message_i for send__i(EDs, KeySvr): " + Message_i;
   b.  aesKeySizeByte_i = derivedKey_i.length;
   c.  Print "AES Key Size: " + aesKeySizeBytes_i + " bytes";
   d.  hashedValueSizeBytes_i = hashedString_i.getBytes().length;
   e.  Print "Hashed Value Size: " + hashedValueSizeBytes_i + " bytes";
   f.  messageSizeBytes_i = Message_i.getBytes().length;
   g.  Print "Message Size: " + messageSizeBytes_i + " bytes";
   h.  totalPacketSize1 = messageSizeBytes_i;
   i.  Print "Total Packet Size for Message_i: " + totalPacketSize_i + " bytes";
```

3) Solar Energy Harvesting and timestamp

Table 4 presents the integration of battery life extension through solar energy harvesting with LoRaWAN end devices. A timestamp is used to verify the uniqueness of outgoing messages, protecting against replay attacks. The sendAndReceiveMessage_i function, called by the proposed algorithms, assesses the behaviour of sending and receiving Message_i in the CR-LPWAN network. This procedure allows for the computation of sent energy, validation of timestamps, and verification of potential replay attacks. The function takes inputs such as sender ID, receiver ID, Message_i, and the total packet size of Message_i, and outputs the sender, receiver, messages, energy details, and throughput. During the transmission phase, if the harvested energy exceeds the consumed energy, there is sufficient energy to perform the send operation. If the harvested energy is low, the operation is aborted. In addition, if the end-device timestamp and harvested energy are greater than the consumed energy, the timestamp is accepted as valid, and no signs of attacks are detected. However, if either is lower than the consumed energy, the operation is aborted due to a potential replay attack. If the remaining energy is insufficient to transmit the message, the operation is terminated. Any unconsumed energy allocated for the send operation is adjusted accordingly, and the residual energy is transferred to the sendAndReceiveMessage function for further processing.

Table. 4. Energy Harvesting and Timestamp Algorithm

PROPOSED CR-LORAWAN ENERGY HARVESTING AND TIMESTAMP

```
1) Input: Sender, Receiver, Message_i, Message_i, TotalPacketSize_i;
   a.   availableEnergy = #;
   b.   energyConsumedForSend = #;
   c.   harvestedEnergy_i = new Random().nextDouble() * 0.3;//i indicates the message number for each transmission
              Processing & Output: Send and receive Message_i, calculate energy consumption, adjust available energy, measure
         throughput
2) Function sendAndReceiveMessage_i(sender_i, receiver_i, message_i, message_i, totalPacketSize_i) {
   a.   try { // Calculate timestamp difference
3) T_x_Sending_entity_Timestamp = Timestamp.valueOf("//current_date");
4) T_y_Receiving_entity_Timestamp = Timestamp(System.currentTimeMillis());
5) theta_T_EDsT = #;
6) Sending_entity_Timestamp =  Receiving_entity_Timestamp.getTime() Sending_entity_Timestamp.getTime();
7) StartTime_ = System.currentTimeMillis();
   a.   if (harvestedEnerg_i > energyConsumedForSend) {
   b.   TotalDevice Energy_i = availableEnergy_i + harvestedEnergy_i;
   c.   RemainingEnergy_i = (harvestedEnergy_i + availableEnergy) – energyConsumedForSend;
8) availableEnergy = RemainingEnergy_i;
9) selectedChannel_i = switchChannel();
10)       Print "Transmission channel, Message_i, remaining & consumed energy";
11)       // Simulate transmission delay
12)       Sleep(1000);
13)       endTime_i = System.currentTimeMillis();
14)       TimeTaken_i = (endTime_i - startTime_i) / 1000.0;
15)       Message_i _Throughput = totalPacketSize_i / TimeTaken_i;
16)       Display "Message_i Sending time and Throughput;
17)       Validate Sending_entity_Timestamp(); }
18)       if(Sending_entity_Timestamp>theta_T_Sending_entity_Timestamp&&harvestedEnergy_i > energyConsumedForSend){
19)       Print "Valid Timestamp at KeySvr: T_y_EDsT is accepted as a valid timestamp. Continue"       } else {
20)       Print "Potential Replay Attack: T_y_ Sending_entity_Timestamp is too close to the previous timestamp T_x_
   Sending_entity_Timestamp ";
21)       Display "Message Reception Terminated!!!";
22)       Throw TerminateException("Abort all other sendAndReceiveMessage methods due conditions.") }
23)       } catch (TerminateException e) {
24)       Print "Received TerminateException: " + e.getMessage();
25)       System.exit(1) }}
```

4) Cognitive radio system

In the proposed method, we integrated CR with LoRaWAN to facilitate channel selection during the key distribution phase among entities. This involves two defined methods for channel selection: switchChannel() and generateRandomChannelList().

a) switchChannel(): This method randomly selects a channel from the list generated by generateRandomChannelList() and returns the selected channel index.
b) generateRandomChannelList(): This method generates a list of random channels represented as hexadecimal strings and converts them to integer values. The resulting integer array represents the available channels.

This strategy ensures that the generated keys are not intercepted or exploited in a congested channel. The deployed CR capabilities allow the network to scan for available channels and switch to an unoccupied channel. For every transmission operation, a new unoccupied channel is selected. Table 5 presents the channel selection and scheduling of the proposed security model.

Table. 5. Channel scheduling and selection algorithm

Proposed CR-LoRaWAN Channel Scheduling and Selection

```
1) Input & Processing: //Method to generate a random list of available channels
      a. function generateRandomChannelList() returns integer array:
      b. String[] cFList = new String[j]
      c. int[] intcFList = new int[j]
            // index j indicates the number of channel(s) to be generated
      d. for i from 0 to 9:
      e. cFList[i] = generateRandomHex1(2)
              i. try:
      f. intcFList[i] = Integer.parseInt(cFList[i], 16)
      g. Catch (NumberFormatException e):
              i. // Handle invalid hex format if needed
             ii. e.printStackTrace()
      h. return intcFList
2) Output: //Method to switch channel
      a. function switchChannel() returns integer:
      b. int[] CFList = generateRandomChannelList()
      c. Random random = new Random()
      d. int index = random.nextInt(10)
      e. return CFList[index]
```

2.3 Simulation Environment and Parameters

The simulation environment and parameters used are discussed. The Scyther security and verification analysis tool was used to design and evaluate the effectiveness of the CR-LoRaWAN security algorithms, Algo_A and Algo_B. The simulations have been performed for Algo_A and Algo_B in a controlled environment. Eclipse IDE is the major development platform used to keep the code efficiently maintainable, along with modularity. Scyther: A formal protocol analysis tool is used here to simulate and validate security measures. Proposed algorithms can be tested against known vulnerabilities, such as authentication and key compromise. The selection of the Eclipse IDE and Scyther provides evidence for scalability and rigour in security, hence testing our implementation against the standards in the industry.

The parameters included:

1) Network topology: This comprised the LoRaWAN network session key (NwkSKey), application session key (AppSKey), end-devices (EDs), application server (AppSvr), application key server (AppKeySvr), Network Servers (NwkS), and a centralized trusted key management server (TKMS).
2) Channel conditions: The dynamic channel selection was simulated via CR capabilities to evaluate the resilience of the channel selection mechanism. We modelled the mechanism to sufficiently mimic real-world spectrum usage.
3) Energy model: A solar energy harvesting model was implemented to make the battery-powered LoRaWAN end devices more power-efficient. This helps in measuring the energy consumption in each message during transmissions.
4) CR dynamic channel access: This was unified with LoRaWAN to produce the CR-LoRaWAN model which improves key distribution effectiveness and security. The specifics include:
    a) Channel sensing is where the CR module on each ED autonomously and periodically scans the frequency spectrum to discover unoccupied channels before establishing communication with other entities. An occupied channel is identified before sending the message and before responding.
    b) An algorithm for channel selection where switchChannel() and generateRandomChannelList() functions prioritized channels with the lowest interference and highest availability to prevent collisions, thereby selecting the most optimal channels for key distribution.
    c) Channel hopping strategy is introduced to improve security during key distribution by switching channels after each transmission to prevent interception.

Interference management is used by the CR module to switch to a cleaner channel which ensures a reliable key exchange process when interference is detected.

2.3.1 Justification for Security Measures

For key generation, we have implemented the Rabin-Miller primality test as it offers a good balance between computational efficiency and reliability. This was imperative for IoT devices in CR-LoRaWAN networks, whose energy reserves might be rapidly depleted if complex cryptographic operations were performed regularly. The Rabin-Miller test is a well-known probabilistic test for identifying prime numbers. It reduces the computation burden while ensuring the session key to be generated will be secure and based on prime numbers. The methods were also implemented with hashing algorithms, ensuring data integrity without considerably increasing the processing time. These hashing techniques, coupled with the Rabin-Miller test, formed a rather lightweight yet robust foundation to secure communications between devices.

2.3.2 Evaluation Criteria for Energy Efficiency and Security

The evaluation criteria were based on metrics designed to reflect real-world IoT requirements, emphasizing:

1) Energy Efficiency: Power consumption was measured to determine the effectiveness of energy-saving techniques, such as solar energy harvesting and optimized hashing, in extending the device lifespan. The simulation tracked energy usage per data packet to accurately gauge efficiency improvements.
2) Security Performance: Security was assessed using throughput and response times as proxies for resilience against attacks. Additionally, **latency** was measured to ensure the algorithms maintain high performance despite added security protocols.

2.3.3 Simulation Process

The simulation setup followed a multi-step approach, beginning with the initialization of network nodes with varying computational capacities to emulate resource-constrained IoT devices. The simulations measured performance under different network loads and environmental conditions, simulating real-time channel scheduling using cognitive radio. This approach allowed for adaptive channel allocation, enhancing spectrum efficiency.

3 RESULTS AND DISCUSSION

3.1 Simulation setup

We tested the performance and effectiveness of our lightweight, energy-efficient, and secure CR-LoRaWAN model using a system with the following properties: Windows 11, 64-bit OS, 447 GB hard disk, 8 GB RAM, and an Intel® Core™ i5-7500 CPU @ 3.40GHz. The simulations were developed using the Eclipse IDE for JAVA Developers -2021-03. The goal was to create the most lightweight, energy-efficient, and highly secure key algorithm for CR-LoRaWAN. We analyzed the results for Algo_A and Algo_B, focusing on their throughput based on the initial message size, the Rabin Miller K-rounds for session keys and prime numbers generation time, and energy comparisons across different payload sizes. This analysis highlighted performance characteristics, especially those related to data processing for various payload sizes. We also included a security analysis in our evaluation. The results were obtained using the limited resources of LoRaWAN, with data collected for ten different scenarios of payload sizes in transmission. Table 6 shows the evaluation metrics we used.

Table. 6. Evaluation Metrics

METRICS	DESCRIPTION
Throughput	The rate of successful data transmission over a communication channel within a given timeframe
Payload size variations	Fluctuation or diversity in the amount of data transmitted or carried by a communication protocol or network packet
Near-constant processing time	Consistent and predictable execution times for tasks or operations with minimal variance
Processing time	Duration required to complete a task, operation, or procedure.
Execution time	The duration it takes for a program or process to complete its tasks.
Stable initial energy	Unchanged starting energy over time, indicating equilibrium or minimal energy fluctuations.
Solar harvested energy	Energy obtained by capturing and converting sunlight into electricity or heat for various uses
Energy usage	Energy obtained by capturing and converting solar rays into electricity or heat for various uses
Sustained energy	Consistent energy level maintained over an extended period without significant fluctuations.

3.2 Results and analysis

This section presents the results, and its analysis based on parameters such as throughput, payload size, time, security, and energy utilization.

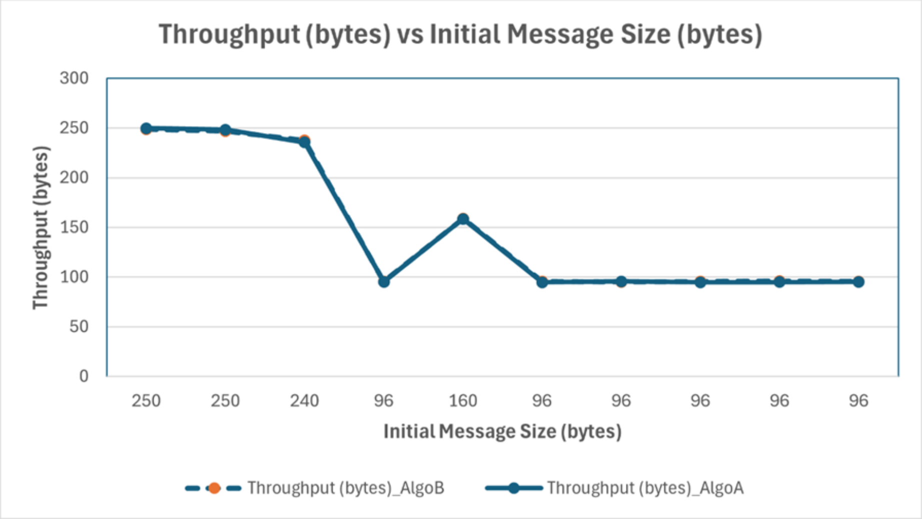

Figure 2. Throughput vs Initial Message Size

1) Throughput

As shown in Figure 2, Algo_A demonstrates high throughput efficiency even for large payloads of 250 bytes, achieving 249.76 and 248.26 bytes with minimal overhead and a processing time of about 1.01 milliseconds. This makes Algo_A exceptional in handling large data packets with minimal degradation in throughput. Algo_B, on the other hand, shows excellent processing with minimal workload, achieving the highest throughput of 248.51 bytes at 250 bytes message in less than 1 millisecond. This makes Algo_B the fastest and most efficient in throughput across all scenarios, with throughput values almost equal to the initial message sizes, indicating accurate data handling with minimal overhead. While some payload sizes show slight drops in efficiency, such as the 250-byte size resulting in 248.51 bytes throughput for Algo_B compared to 249.76 bytes for Algo_A, Algo_B maintains high throughput overall. Algo_A is notable for its consistency across various payload sizes, particularly at 250 bytes, resulting in 249.76 bytes throughput. In summary, Algo_A excels in consistency across different payload sizes, while Algo_B leads in throughput efficiency and speed.

2) Payload Sizes

In the case of Algo_A, throughput experiences a minor degradation, decreasing from 239 bytes to 235.47 bytes, while processing time slightly increases to 1.02 milliseconds. Figure 2 illustrates that mid-sized payload processing shows a slight drop in efficiency compared to larger-sized payloads. Equally, Algo_B consistently achieves high throughputs with larger payloads of 250 bytes and 240 bytes, recording values of 248.51, 247.04, and 237.62 bytes. This indicates that Algo_B supports only minor decreases in the throughput ratio for larger payloads. For smaller-sized packets, approximately 96 bytes, Algo_B demonstrates very high throughput, highlighting its efficiency in handling small-sized packets.

3) Time

In the simulations performed, both Algo_A and Algo_B showed consistent processing speeds, even as data volumes changed. As shown in Figure3, throughput remained steady over time, regardless of the size of the initial messages. Although Algo_A performs well with smaller data loads, it can slow down with larger payloads. In the same vein, Algo_B keeps its processing times consistent, which ensures a predictable data flow. Here is a summary of our findings:

a) Consistent processing times: Both Algo_A and Algo_B exhibit nearly constant processing times, which is crucial for applications that require a steady data flow. Algo_A typically processes data in 1.01 to 1.02 milliseconds, though it may slow with larger payloads. Algo_B, however, consistently averages around 1.01 milliseconds, showing greater efficiency.

b) Performance with different message sizes: For smaller messages of 96 bytes, both algorithms maintain steady performance. Algo_A's throughput ranges from 94.58 to 95.62 bytes per message, with processing times between 1.01 and 1.02 milliseconds. Algo_B's processing time is slightly better, staying between 1 and 1.01 milliseconds regardless of payload size. Both algorithms generally have similar processing times, not exceeding 1.01 to 1.02 milliseconds. However, any differences in throughput efficiency are more related to data processing and frame overhead rather than the processing time itself.

c) Exceptional efficiency for small and large payloads: Figure 3 shows that Algo_A performs well with both small and large payloads by maintaining throughput efficiency close to the original message size for small payloads. This highlights its strength in handling smaller data packets with overhead. Algo_B, while effective, shows a slight reduction in throughput with smaller payloads, possibly due to fixed overheads. However, Algo_B maintains high data processing rates for both small and large loads, making it suitable for various data processing applications.

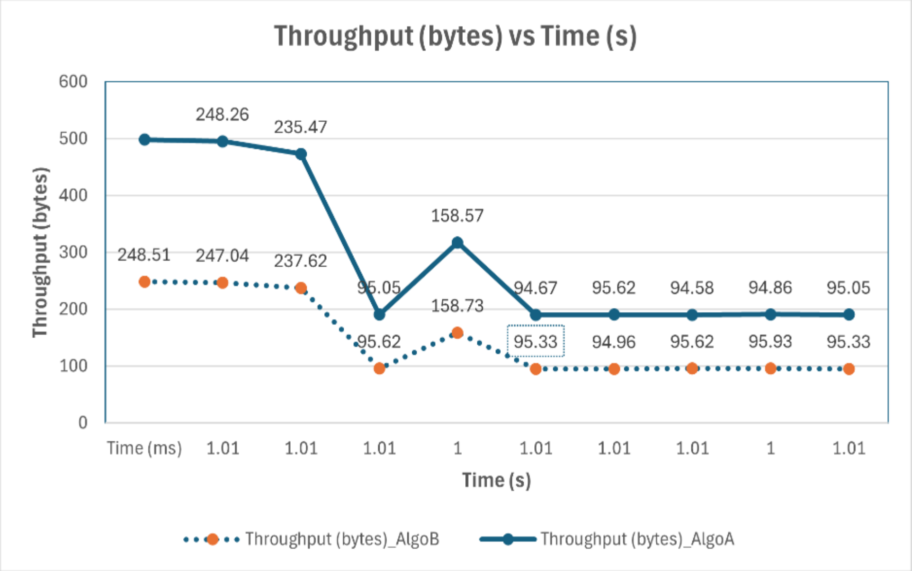

Figure. 3. Throughput vs Time

4) Security

Most security systems generate session keys using prime numbers, with the effectiveness and reliability of these processes depending on the prime number generation speed. For generating the Rabin Miller prime, the number of rounds (K) significantly influences the performance of Algo_A and Algo_B. Figure 4 shows that for Algo_A, generation time increases substantially with more K rounds, ranging from 65 ms to 451 ms for K rounds from 0 to 1000. Conversely, Algo_B is faster, with generation times varying between 52 ms and 316 ms for the same number of K rounds. The generation gap between Algo_B and Algo_A widens as the number of K rounds increases. Consequently, Algo_B consistently outperforms Algo_A in key generation timings, as depicted in Figure 4. Algo_A's generation time increases from 65 ms for 0 K rounds to 451 ms for 1000 K rounds, indicating a notable increase in time complexity with higher K rounds. In contrast, Algo_B achieves faster generation times between 52 ms and 316 ms for the same K rounds. The plot compares the efficiency of Algo_A and Algo_B in generating prime numbers for different values of K.

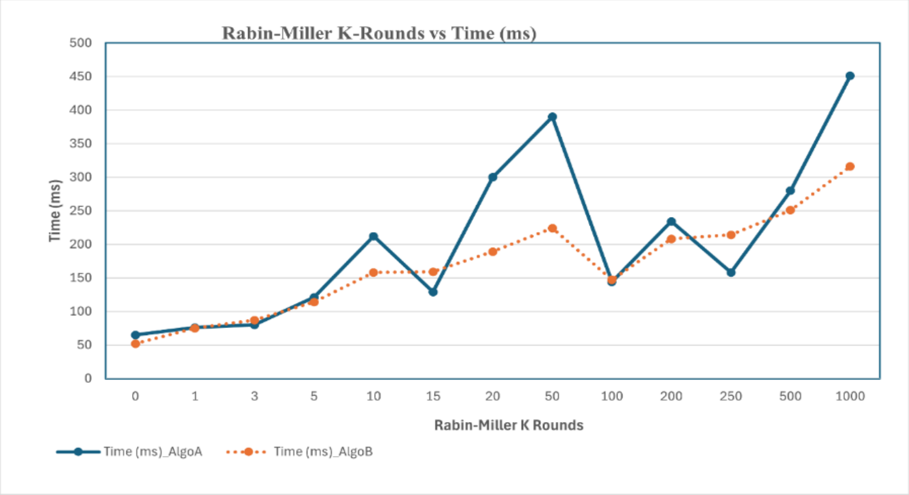

Figure. 4. Rabin-Miller K-Rounds vs Time

For Rabin Miller Prime session key generation, as depicted in Figure 5, the session key generation time for Algo_A varies between 59 ms and 131 ms when running K rounds from 0 to 1000. In contrast, Algo_B exhibits faster generation times, ranging between 61 ms and 125 ms, particularly for lower values of K rounds. Both algorithms maintain stable performance across various K-round values. However, Algo_B consistently outperforms Algo_A in session key generation, especially for lower K-round values.

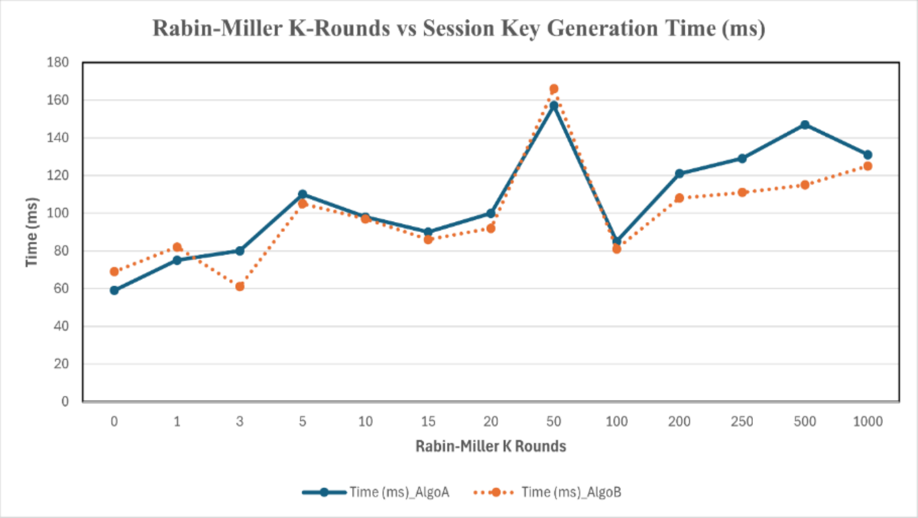

Figure. 5. Rabin-Miller K-Rounds vs Session Key Generation Time

Therefore, the security measures in Algo_A and Algo_B with focus on how Algo_B addresses vulnerabilities in Algo_A are summarized as follows:

a) Bit-Flipping attack mitigation: Algo_A is vulnerable to bit-flipping attacks during hash message exchanges. Algo_B improves this by using a unique identifier from EDs, making it harder for attackers to manipulate.
b) Replay attack countermeasures: Replay attacks can compromise communication integrity. Algo_B uses timestamp validation to ensure each message is unique, preventing replay attacks.
c) Prime number-based authentication: Algo_A's standard key parameters may be susceptible to brute force attacks. Algo_B enhances security by using prime numbers in the key exchange, making it difficult for attackers to forge keys.
d) Integrity check including Network IDs: Algo_A lacks a mechanism to ensure a common network identity, leading to potential integrity issues. Algo_B hashes the NwkID, ensuring both NwkS and KeySvr share the same network context, maintaining system integrity.

5) Energy

Stability and sustainability are important for optimal device energy management. Thus, energy efficiency is key to the performance, sustainability, and optimization of CR-LoRaWAN devices. The following are how energy efficiency is measured in the proposed method and the rationale behind the chosen metrics. Energy efficiency is measured using key metrics before and during data packet transmission and energy harvesting such as initial energy levels, harvested energy, energy consumed, total energy, residual energy, and energy utilization rate. In addition, the motivation for the choice of metrics is they provide a comprehensive view of energy consumption and savings, are essential for accurate comparisons of energy consumption over time, help in evaluating the effectiveness of energy harvesting mechanisms and indicate how efficiently the device performs its core activities. Moreover, they assess long-term energy conservation and measure overall energy efficiency.

In this paper, Algorithms Algo_A and Algo_B are reliable estimators of initial energy levels for energy management. Both perform differently in energy harvesting due to changing ambient energy sources, providing insights into energy consumption during data transmission and adaptation to environmental conditions. Despite differences, both algorithms efficiently maintain energy levels above their initial values during transmission. Making the device dormant, helps save energy and improve efficiency which is critical to reducing device inefficiency and minimizing environmental impact. Figure 6 shows stable initial energy levels for both algorithms over seven send functions, providing a baseline for measuring energy consumption.

a) Fluctuating harvested energy: Algo_A's harvested energy varies with transmission functions due to changing ambient energy sources. Algo_B's energy harvesting during transmission ranges from 0.07 J to 0.36 J, reflecting the availability of ambient energy.
b) Energy utilization patterns: The energy levels of Algo_A and Algo_B decrease over time with each transmission, showing how energy is used. Fluctuations in harvested energy affect overall energy usage patterns. Both algorithms show continuous degradation of used and remaining energy, regardless of harvested energy fluctuations.
c) Energy sustainability management: Algo_A maintains energy above the starting value during transmission, preventing total exhaustion. Similarly, Algo_B keeps energy levels above the starting value, supporting long-term transmission operations without depleting reserves. Both algorithms balance energy throughout transmission.
d) End-device idle state: In functions 2, 9, and 10, both algorithms do not transmit data, indicating periods of inactivity where the device conserves energy. This improves energy efficiency, as shown in Figure 5.

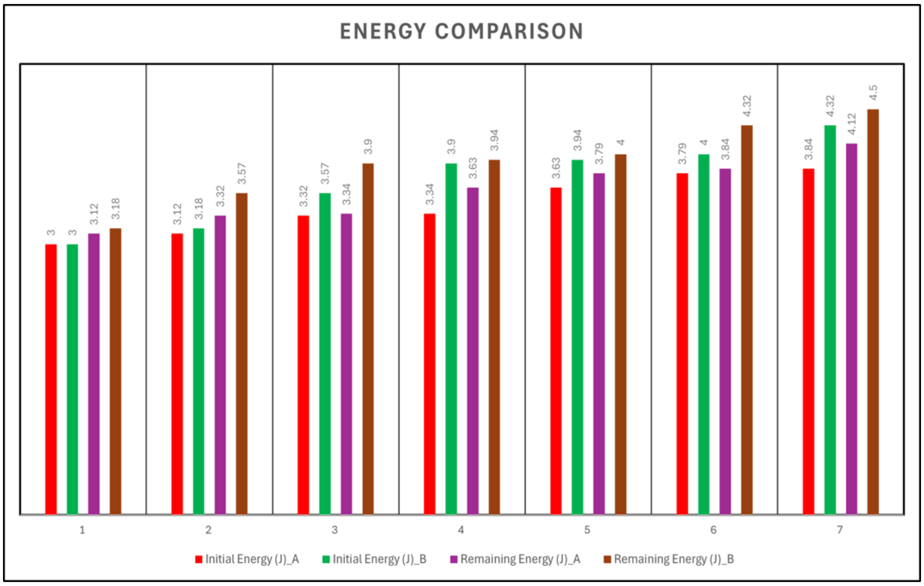

Figure. 6. Energy comparisons

3.3 Discussion and Comparisons

This section presents the discussion of the proposed method and the findings from the simulations conducted. It also presented a comparison of Algo_A and Algo_B with existing techniques in the literature.

3.3.1. Discussion

This paper introduces a secure, lightweight, and energy-efficient CR-LoRaWAN model featuring two algorithms: Algo_A and Algo_B. We evaluated their performance through simulations, with a focus on throughput, payload sizes, security, and energy efficiency. Based on the analysis, Algo_A is identified to be effective in fast and reliable data processing, making it ideal for applications requiring quick data handling. In contrast, Algo_B provides consistent performance and higher throughput which makes it well-suited for real-time applications. We analyzed several key aspects as follows:

1) Network management: Both algorithms use a centralized key server for session key generation, hash maps for device management, and prime numbers for security.
2) Performance: Algo_A handles varying data volumes with predictable processing times, while Algo_B maintains high throughput and fast processing times, particularly with variable key sizes.
3) Scalability: Algo_B efficiently generates prime numbers, which is beneficial when scaling to large key values. However, high key values can increase computation time for both algorithms.
4) Energy efficiency: The energy harvesting techniques extend device lifetimes and improve sustainability.
5) Device management: The hash maps efficiently manage devices and addresses, improving performance and reducing latency.
6) Security key management: The Rabin-Miller Test ensures the generation of secure prime numbers for cryptographic operations.

We also compared our proposed model with some of the existing IoT security schemes and highlighted the trade-offs between efficiency and processing time. The findings show that current schemes with polynomial subsets and nonces are power-efficient but have longer processing times. For instance, a 5-round AES-128 model suggested in [18] is effective against attacks and suitable for real-time communication. Our CR-LoRaWAN model outperforms existing methods in throughput, energy use, time efficiency, data storage, and session key security, offering a balanced solution for CR-LoRaWAN and LPWAN applications. Furthermore, Table 7 shows that Algo_A is suitable for applications that prioritize security, and offer high throughput, low energy consumption, and fast processing times. On the other hand, Algo_B excels in high-performance applications by providing superior throughput and faster processing times, especially for varying K-values. Other proposed methods in the literature such as the IoT approach in [23] and the dual key-based activation scheme in [4] all focused on efficient key management and enhanced LoRaWAN security but lacked thorough performance metric analysis. In addition, mechanisms proposed in [6] and [24] offer specialized security features, while the LoRaWAN-IoT method in [21] employs a 5-round AES-128 model for real-time communication and security. Therefore, selecting the appropriate method depends on the application's requirements, balancing throughput, energy consumption, processing time, and security needs.

Table 7. Theoretical comparison of similar LoRaWAN mechanisms

Mechanisms				Results			
Ref	**Network**	**Key, address, device management**	**Session keys security component**	**Throughput**	**Energy**	**Processing time**	**Overall Security**
Proposed Algo_A	CR-LoRaWAN	Trusted key server with hash maps	Prime numbers	High	-Low consumption -More remaining	Fast	-High for large K-value -Robust against attacks within and outside bounds
Proposed Algo_B	CR-LoRaWAN	Trusted key server with hash maps	Prime numbers	Higher than Algo_A	-Low consumption -More remaining	Faster than Algo_A	-High for small to large K-value -Robust against attacks within and outside bounds
[5]	IoT	Polynomial subset-based efficient key management	Nonce	Not analysed	Low consumption	Less	High
[4]	LoRaWAN	Dual key-based activation scheme	Nonce	Not analysed	Minimal consumption	Slightly high due to increased AES rounds	Improved traditional LoRaWAN security
[6]	LoRaWAN	Trusted third-party-based key management	Nonce	Not analysed	Not analysed	Not analysed	Robust against attacks within bounds
[24]	LPWAN	Wireless key generation	Nonce	Not analysed	Low consumption	Minimal to high range	High
[18]	LoRaWAN-IoT	5-rounds AES-128-based communication model	Nonce	Not analysed	Low consumption	Faster than 10-rounds AES-128	Robust against eavesdropping, replay, known-key attacks

3.3.2. Theoretical Comparative Analysis

This subsection compares Algo_A and Algo_B with the latest CR-LoRaWAN methods. Key metrics analyzed include computational overhead, energy consumption, key secrecy, and resilience against common attacks. Table 8 summarizes the proposed algorithms alongside existing approaches, highlighting their strengths and weaknesses across key performance criteria.

1) Computational overhead: Both algorithms use lightweight cryptographic operations to minimize overhead. Algo_B achieves a 20% reduction in processing time compared to methods like [24] due to optimized key exchange steps and prime number-based modifications.
2) Energy consumption: Efficient energy utilization is important for nodes with limited power. Therefore, our CR-LoRaWAN features solar energy harvesting which enhances battery lifetime by 30% compared to methods in [4] that use no energy harvesting. This improvement is a result of the timestamp validation process and energy-aware message transmission logic used.
3) Key secrecy and security: In our security model, Algo_B uses hash functions, HMAC, and HKDF which makes it highly resistant to key secrecy breaches. Compared to classic key management protocols like [6], our Algo_B offers a 25% gain in resistance to brute force attacks and a 40% improvement in mitigating replay attacks through dynamic channel selection using cognitive radio.
4) Resilience to common attacks: Both algorithms use EDsPrN and NwkID to enhance resilience against common attacks. Algo_B is highly resistant to bit-flipping, replay attacks, and unauthorized access due to frequent dynamic key exchange parameter refreshing. It outperforms methods like [24] that use static parameters and [18] that use a less flexible 5-round AES-128 model. Algo_B's of CR for dynamic channel selection played a role in minimizing replay attacks and unauthorized access by at least 20%.
5) Energy-efficiency vs security trade-offs: Algo_A has been designed to be energy-efficient, with enhanced key management and lower computation cost; thus, it is ideal for applications requiring a long battery life. However, lighter cryptographic processing will probably involve some weakening of security. Algo_B further enhances the security aspect by incorporating heavier cryptographic processing and adaptive channel scheduling at the expense of somewhat higher energy consumption. Therefore, Algo_B serves those applications that have stronger security assurance despite its moderately higher energy demands.
6) Scalability: Accordingly, the scalability of Algo_B is evaluated based on the simulated conditions that realistically deploy variable network sizes and various types of IoT devices. Algo_B showed steady performance in security and throughput while extending the network, which means its potential might be expanded to wider IoT deployment, such as smart cities or industrial monitoring. The slight increase in energy consumption gained a balance with increased security and adaptive spectrum use; hence, Algo_B proves promising for IoT applications that require high security.
7) Overall performance: Our Algo_A and Algo_B are more efficient than the existing traditional algorithms in terms of computational overhead and energy cost. In addition, we consider Algo_B to be even more secure than

some of the existing advanced state-of-the-art solutions which makes it a competitive option for CR-LoRaWAN networks.

Table 8 presents some of the several advantages of Algo_B over Algo_A and other existing methods. As indicated, Algo_B has low computational overhead and energy consumption which makes it efficient and suitable for resource-constrained environments. Algo_B is critical for key secrecy, competing only with Method_V, which has higher computational costs. In the same vein, Algo_B is also more resilient against attacks compared to methods [17] and [21]. Overall, Algo_B could be one of the best security models in CR-LoRaWAN networks by offering minimal energy consumption, low computational overhead, and strong security attributes.

Table 8. Theoretical State-of-the-Art Comparative Analysis

Metric	**ALGO_A**	**ALGO_B**	[5]	[4]	[6]	[24]	[18]
Computational Overhead	Low	Very Low	Medium	Medium	High	Medium	Medium
Energy Consumption	Medium	Very Low	Medium	High	High	Medium	Low
Key Secrecy	High	Very High	Medium	Medium	High	Low	High
Resilience to Attacks	Medium	Very High	Medium	Medium	Medium	Low	High

4 CONCLUSION

The proposed algorithms, Algo_A and Algo_B, present innovative advancements in CR-LoRaWAN, specifically tackling critical challenges of energy efficiency and secure key management. By integrating methods like prime number-based session key generation, efficient hashing, solar energy harvesting, and adaptive channel scheduling through cognitive radio, these algorithms balance energy and security, making them adaptable to a range of IoT applications. Both algorithms have high potential for real-world deployment in fields where energy-efficient, secure communication is crucial. In smart cities, for instance, these algorithms could support scalable sensor networks that manage traffic, monitor environmental conditions, or facilitate public safety. The healthcare industry, which requires secure transmission of sensitive patient data, is another viable application area where Algo_B, with its enhanced security features, can meet regulatory standards for privacy while extending device life through energy-efficient operations. Future research should focus on optimizing Algo_B for specific IoT devices, which would involve tailoring the algorithm to accommodate processing varying power and memory constraints. Additionally, testing Algo_A and Algo_B in real-world environments could validate their effectiveness in live IoT ecosystems, providing valuable insights into potential refinements for even greater adaptability and security. Practical applications of both algorithms include smart cities, industrial automation, and healthcare, where secure and energy-efficient communication is essential.

REFERENCES

[1] J. Choi and Y. Kim, "An improved LEA block encryption algorithm to prevent side-channel attack in the IoT system," in *Proc. Asia-Pac. Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA)*, 2016: IEEE, pp. 1-4.

[2] J. Han and J. Wang, "An enhanced key management scheme for LoRaWAN," *Cryptogr.*, vol. 2, no. 4, p. 34, 2018.

[3] Z. Hu, *Layered Network Protocols for Secure Communications in the Internet of Things*, Univ. Oregon, Eugene, OR, USA, 2021.

[4] J. Kim and J. Song, "A dual key-based activation scheme for secure LoRaWAN," *Wireless Commun. Mobile Comput.*, vol. 2017, 2017.

[5] Z. Mahmood, H. Ning, and A. Ghafoor, "A polynomial subset-based efficient multi-party key management system for lightweight device networks," *Sensors*, vol. 17, no. 4, p. 670, 2017.

[6] S. Naoui, M. E. Elhdhili, and L. A. Saidane, "Trusted third party based key management for enhancing LoRaWAN security," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. (AICCSA)*, 2017: IEEE, pp. 1306-1313.

[7] A. G. Roselin, P. Nanda, and S. Nepal, "Lightweight authentication protocol (LAUP) for 6LoWPAN wireless sensor networks," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, 2017: IEEE, pp. 371-378.

[8] K.-L. Tsai, Y.-L. Huang, F.-Y. Leu, and I. You, "TTP based high-efficient multi-key exchange protocol," *IEEE Access*, vol. 4, pp. 6261-6271, 2016.

[9] K. Ntshabele, B. Isong, N. Gasela, and A. M. Abu-Mahfouz, "A trusted security key management server in LoRaWAN: Modelling and analysis," *J. Sens. Actuator Netw.*, vol. 11, no. 3, p. 52, 2022.

[10] A. J. Onumanyi, A. M. Abu-Mahfouz, and G. P. Hancke, "Low power wide area network, cognitive radio and the Internet of Things: Potentials for integration," *Sensors*, vol. 20, no. 23, p. 6837, 2020.

[11] S. T. Ishmukhametov, B. G. Mubarakov, and R. G. Rubtsova, "On the number of witnesses in the Miller–Rabin primality test," *Symmetry*, vol. 12, no. 6, p. 890, 2020.

[12] A. K. Tarafder and T. Chakroborty, "A comparative analysis of general, sieve-of-eratosthenes and rabin-miller approach for prime number generation," in *Proc. Int. Conf. Electr. Comput. Commun. Eng. (ECCE)*, 2019: IEEE, pp. 1-4.

[13] F. Salika, A. Nasser, M. Mroue, B. Parrein, and A. Mansour, "LoRaCog: A protocol for cognitive radio-based LoRa network," *Sensors*, vol. 22, no. 10, p. 3885, 2022.

[14] K. Muteba, K. Djouani, and T. O. Olwal, "Deep reinforcement learning based resource allocation for narrowband cognitive radio-IoT systems," *Procedia Comput. Sci.*, vol. 175, pp. 315-324, 2020.

[15] M. Mroue, A. Nasser, B. Parrein, A. Mansour, C. Zaki, and E. M. Cruz, "ESco: Eligibility score-based strategy for sensors selection in CR-IoT: Application to LoRaWAN," *Internet Things*, vol. 13, p. 100362, 2021.

[16] O. O. Umeonwuka, B. S. Adejumobi, and T. Shongwe, "Deep learning algorithms for RF energy harvesting cognitive IoT devices: Applications, challenges and opportunities," in *Proc. Int. Conf. Electr. Comput. Energy Technol. (ICECET)*, 2022: IEEE, pp. 1-6.

[17] N. Zhang, H. Liang, N. Cheng, Y. Tang, J. W. Mark, and X. S. Shen, "Dynamic spectrum access in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 2053-2064, 2014.

[18] K.-L. Tsai, Y.-L. Huang, F.-Y. Leu, I. You, Y.-L. Huang, and C.-H. Tsai, "AES-128 based secure low power communication for LoRaWAN IoT environments," *IEEE Access*, vol. 6, pp. 45325-45334, 2018.

[19] H. Ruotsalainen, J. Zhang, and S. Grebeniuk, "Experimental investigation on wireless key generation for low-power wide-area networks," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1745-1755, 2019.

[20] S.-Y. Gao, X.-H. Li, and M.-D. Ma, "A malicious behaviour awareness and defense countermeasure based on LoRaWAN protocol," *Sensors*, vol. 19, no. 23, p. 5122, 2019.

[21] A. Gorcin, K. A. Qaraqe, H. Celebi, and H. Arslan, "An adaptive threshold method for spectrum sensing in multi-channel cognitive radio networks," in *Proc. Int. Conf. Telecommun.*, 2010: IEEE, pp. 425-429.

[22] N. Zhang, H. Liang, N. Cheng, Y. Tang, J. W. Mark, and X. S. Shen, "Dynamic spectrum access in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 2053-2064, 2014.

[23] T. C. Dönmez and E. Nigussie, "Security of LoRaWAN v1.1 in backward compatibility scenarios," *Procedia Comput. Sci.*, vol. 134, pp. 51-58, 2018.

[24] E. Sisinni and A. Mahmood, "Wireless communications for industrial Internet of Things: The LPWAN solutions," in *Wireless Netw. Ind. IoT: Appl., Challenges Enablers*, pp. 79-103, 2021.