Analysis the Effect of Dynamic Clustering and Lightweight Symmetric Encryption Approaches on Network Lifetime in WSNs

Tarik Abu-Ain
Saudi Electronic University
RAMI AHMAD (r_a_sh2001@yahoo.com)
Sebha University https://orcid.org/0000-0003-3913-6397
Elankovan A Sundararajan
UKM: Universiti Kebangsaan Malaysia

Research Article

Keywords: Lightweight encryption, Wireless sensor networks, Symmetric encryption, WSN security, WSN Lifetime, IoT Security.

Posted Date: May 18th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-446269/v1

Abstract

Energy consumption and security efficiency are still main challenges in Wireless Sensor Networks (WSNs) due to their hardware resource-constrained nature. The 6LoWPAN protocol was developed to improve WSNs communication, security, and node management optimization. Hence the protocol energy efficiency and security can be improved. In this paper, we address the WSN nodes' power consumption by analyzing the Dynamic Cluster Head (DynCH) technique, which automates the process of selecting WSN Cluster Head (CH) nodes based on the WSN nodes' energy and nodes' distances among each other in mobile WSN nodes. Moreover, this analysis covers the complexity of DynCH in different environments to prove its efficiency compared to the steady CH mechanism. In addition, we present the performance analysis of different lightweight systematic block encryption algorithms along with DynCH scheme on WSNs lifetime. In particular, Speck128, FlexenTech, Tiny Encryption Algorithm (TEA), and Advanced Encryption Standard (AES) algorithms are used in order to determine the amount of energy consumed by the sensor nodes and their effect on the network lifetime. Furthermore, the lightweight key management mechanism is used to secure the data and keys exchange between WSN nodes in all different systematic encryption algorithms. The Cooja simulator with Contiki operating system are used to evaluate our analysis. Finally, the outcome of the analysis has showed that DynCH improves the wireless network lifetime by 45% compared to the steady clustering approach. Moreover, the analysis also shows that, Speck128 consumed 26%, FlexenTech consumed 52%, TEA consumed 65%, and AES consumed 78% of wireless network lifetime compared to unsecure wireless networks communication, respectively.

1 Introduction

In recent years, wireless communication plays a vital role in real-life situations. the use of Wireless Sensor Networks (WSNs) has been applied in various fields such as healthcare, environmental, and scientific monitoring in various domains and other applications. A WSN contains several sensor nodes that communicate with each other using radio frequencies. They are able to perform different tasks including environmental sensing, measuring, monitoring and data processing [1]. These WSN nodes are resource-limited devices that have narrow bandwidth, limited battery life, restricted memory capacity and low processing power [2].

6LoWPAN and Zigbee are two widely used protocols in WSNs [3]. Both utilize the IEEE 802.15.4 physical and Media Access Control (MAC) layers, with the difference that the later protocol introduced the IPv6 over low-power wireless personal area networks. 6LoWPAN standard protocol utilizes the Advanced Encryption Standard (AES) to secure the WSN communication [4]. However, the WSN energy consumption and security can be optimized to enhance the network lifetime and improve the Quality of Service (QoS).

Having secure and energy efficient WSN is fundamental to its operation. To achieve that, clustering and encryption are often used. Clustering aims at grouping the nodes in a cluster, where a Cluster Head (CH) node is elected. This node is responsible for gathering the nodes' data, aggregate it, perform some simple operations on it such as compression and de-correlation, and ultimately sending it to a remote node for

further processing and analysis [5]–[9]. Clustering is often used in a stationary environment with static nodes. However, Mobile Wireless Sensor Networks (MWSNs) are widely used in many scenarios and applications where static nodes cannot cover the entire Area of Interest (AoI). Therefore, having a dynamic clustering algorithm is crucial for MWSNs. Many of the available studies [5]–[9] focused on clustering for static WSN nodes, whereas the available dynamic clustering studies have a high degree of complexity [10]–[13]. Hence, the authors in [14] have proposed new Dynamic Clustering (DynCH) scheme to support dynamic clustering in MWSNs in a simple and effective manner. The DynCH scheme was based on the power level of the WSN nodes, the distance among each other, the distance between them and the Edge Router (ER) location, and the minimum number of adjacent WSN nodes that liked with each CH. Moreover, WSNs are often distributed in open areas, where it utilizes open wireless media to transmit its sensed data [15]. Thus making it prone to several attacks and threats. Therefore, securing WSN is paramount to its operation. Encryption is widely used to secure data. For instance, in 6LoWPAN protocol, the AES algorithm is used to encrypt and secure data transmission [4]. Encrypting WSN data prevents it from many well-known attacks since it hides the information by encrypting them, thus making them more robust against any intruder [16]. Knowing that we have to distinguish between intrusion detection algorithms that are used to enhance security and detect enemy information [17], [18], and encrypt data which is part of the confidentiality of information security.

Encryption in WSN should be done efficiently to prolong the network lifetime and consider the limited resources of the sensor nodes. Therefore, proposing an efficient encryption algorithm that meets the WSN requirements is very challenging and an ongoing research problem [19], especially that traditional encryption algorithms are designed for resourceful devices and equipment in terms of processor, power, communication bandwidth and memory and not well optimized for WSN. However, energy limitation and sacristy for WSN is the most challenging design factor when deploying encryption algorithms on these nodes, since having an inefficient encryption algorithm will deplete the node energy and reduces the network lifetime. Sophisticated encryption algorithms are not suitable for WSNs, not only since it needs powerful resources but also its security features and requirement may conflict with the rea-time operation of the sensor node due to its limited computational power and communication capabilities [20].

Lightweight encryption algorithms have been recently proposed as a viable solution for securing WSNs, where a trade-off between the security strength measures and energy efficiency are investigated. On one hand, increasing the security level consumes more energy and results in reducing the network lifetime, and vice versa. These algorithms require less memory, processing power, energy resources and take shorter processing time [21], thus making them feasible for WSNs. The majority of research work in that direction focus on symmetric encryption algorithms due to their low computation complexity when compared with asymmetric encryption algorithms. Hence, one of our contributions of this paper is to study the evaluation of various lightweight symmetric protocols. Mainly, the AES, Speck128, Tiny Encryption Algorithm (TEA) and FlexenTech [22] with all supported key sizes (128, 192 and 256 bits) are analysed and compared. For Speck128, the 128-bit block size configuration was selected since the other block ciphers also use 128 bit blocks. Furthermore, it is known that none of the AES, TEA, Speck128 protocols has been compromised, thus they are considered reliable and secure algorithms [23]. In

addition, key-management, which includes creating, updating, saving and allocating the encryption keys on the sensor nodes, is another crucial process for securing WSN. In symmetric encryption, the secrete key, which is used for both encrypting and decrypting data are used to build the secure communication link between the WSN nodes [24]. Therefore, it is important to protect the key and to exchange it securely within the WSN. In many circumstances, WSN requires security and availability, therefore, this paper analyses the performance of the DynCH algorithm [14] for MWSN while using several lightweight symmetric encryption algorithms to see their effect on the lifetime of WSN networks. A summary of the main paper contributions are as follows:

1. The performance of the DynCH clustering algorithm, specially designed for MWSNs, is analyzed and compared with the static clustering technique. The obtained simulation results show the efficiency of the DynCH algorithm.
2. The performance of using various lightweight symmetric encryption algorithms in WSN nodes is analyzed and compared with each other by influencing the lifetime of WSNs and to determine the best implementation in 6LoWPAN.

The rest of the paper is organized as follows: Sect. 2 presents background information and summarizes the most related work. Section 3 discusses the DynCH dynamic clustering algorithm and the lightweight symmetric encryption algorithms. Section 4 presents the simulation environment, setup, results and the performance metric. Section 5 provides a discussion of the attained simulation results under different environment and using different encryption algorithms, Finally, Sect. 6 concludes the paper and draw future work.

2 Background And Related Works

The purpose of this section is to provide the technical background and literature review of topics of interest to this paper. Particularly, the 6LoWPAN Protocol, clustering and cryptographic algorithms in WSNs.

2.1 6LoWPAN Protocol

6LoWPAN is a protocol built on top of the IEEE 802.15.4 utilizing the 2.4 GHz spectrum and standardized by the Internet Engineering Task Force (IETF), which is designed to support low range, power, memory usage and cost. Thus making it very suitable for WSNs. The prime feature of this protocol comes with its capability to support the IPV6 stack, which makes it capable of connecting to other types of wireless network nodes that support IPv6 such as Bluetooth, WiFi and sub-1 GHz low power radio frequency [25] by using an edge router. Figure 1 depicts a WSN running the 6LoWPAN protocol.

In order to further reduce the WSN running 6LoWPAN protocol, two modifications occurred on the IEEE 802.15.4 protocol, the *e* and *g*, i.e. IEEE 802.15.4*e/g.* Moreover, reliability between WSN has been improved by running a set of rules to manage and control data exchange between the nodes.

Furthermore, the 6LoWPAN physical (perception) layer is responsible for transforming the digital bits into analog wave to be transmitted over the air. Pertaining to the network layer, 6LoWPAN WSN nodes can either be a router or host node. Nodes can explore the neighbouring nodes and establish a network topological graph. It uses the Routing Protocol for Low-Power and Lossy Networks (RPL) to forward packets to other nodes. RPL defines two types of routing modes; storing and non-storing. In the former, all nodes are considered as routers where the nodes; routing tables are populated and exchanged between the nodes. While the non-storing mode has only one router node named the edge-router [26].

The edge router is needed to connect different IP networks together and forwards different packets between different media. For data transmission protocol, both Transmission Control Protocols (TCP) and User Datagram Protocol (UDP) can be used. To secure data transmission over UDP, the Datagram Transport Layer Security (DTLS) protocol can be used on top of UDP [27], while the Transport Layer Security (TLS) and the AES-128 encryption algorithm can run on top of TCP, and used for link layer encryption and authentication. However, TLS/DTLS employment requires special hardware resources such as a hardware encryption engine to be able to use advanced ciphering algorithms [26]. Consequently, deploying a lightweight encryption algorithm is very crucial for WSNs, which have limited hardware resources and are made to be cost-effective and have simple hardware design. Regarding the application later, the Constrained Application Protocol (CoAP) which runs over UDP is often used as a replacement of the HTTP as it requires fewer resources. Another protocol that is similar to CoAP and runs over TCP is the Message Queue Telemetry Transport (MQTT).

2.2 Clustering Technology in WSN Networks

In order to prolong the WSN lifetime and preserve the nodes' energy, clustering is often used, where nodes are grouped into clusters. Each cluster elects a Cluster Head (CH) node, which is responsible for gathering the sensed data from the sensor nodes, aggregating and compressing them, then send them to a remote site for data collection, more advanced processing and decision process [28].

In the literature, several works have been published in the area of WSN clustering and routing. The authors in [7] presented a joint clustering and routing protocol that aims at reducing long-distance communication with the sink node and optimize the power consumption. The AoI is divided into different layers until reaching the sink node. Furthermore, each layer is divided into different identical clusters, whereas the number of clusters increases as becoming closer to the sink node. Two Cluster Heads (CH) are assigned for each cluster in order to balance the energy consumption within the cluster nodes. The first CH is called the leader CH that is responsible for sending the collected data to the sink node, while the other CH is responsible for gathering data from the cluster nodes.

The authors in [6] modified the previous work by proposing to have a forwarding-head node on each layer as a replacement of the leader-head CH for each cluster. However, both proposals suffered load-balancing challenges which were resolved in [29]. In [8], the authors proposed different criteria for selecting the CH such as nodes' remaining energy, number of neighbour nodes and the distance between the nodes' and the sink node (ER). Another work that alternates between the nodes' selection for being a CH is proposed

in [30]. The alternation depends on the system's overall nodes' energy such that the CH selection process is uniform among the nodes, thus prolonging the network lifetime.

Arumugam et al. [31] proposed an energy aware clustering algorithm and routing protocol that aims at optimizing network energy consumption. The CHs for each cluster are chosen for each cluster to minimize the energy consumption and optimize the sensor nodes' resources. The routing protocol chooses the node with the highest residual energy which helps to provide a high packet delivery ratio. Another distinctive clustering strategy was presented in [5], the authors utilized different parameters that are related to the nodes' remaining energy, degrees and centrality factors to select the CH node.

In dynamic clustering technique, the authors in [10] presented a self-organization clustering scheme to maintain relay networks of mobile data collectors. The proposed algorithm distributed the WSN nodes to different logical groups based on the convex hull algorithmic problem to reduce signalling overhead. Moreover, each local group has a locale mobility management. In [11], the authors used a genetic algorithm to choose the CH amongst WSN nodes based on different factors such as WSN node coverage range, the distance between WSN nodes, the ER, and the expected consumed power of WSN node to select CH nodes for both single-hop and multi-hop models. Furthermore, the authors in [12] used K-means to perform the dynamic CH selection. The ER used it to learn the preferences and priorities of a group of WSN nodes to attain optimal solutions over time using the information at the network level, whereas, partial learning enabled each individual WSN node to learn about their preferences and priorities to achieve optimal solutions over time using neighbourhood information. Another work which combines static and dynamic clustering is presented in [13]. The authors condidered a mixture of static and dynamic combinations along with layered-based multi-hop communication. Initially, the ER connects the WSN nodes to different layers based on their distance from the ER which is used for inter-cluster multi-hop connections. It also calculates the adjacent characteristics of the WSN nodes. The CHs are then shaped in a distributed manner, whereby the chance of WSN node becoming a CH depends on its energy level, harvested energy, neighbourhood attributes, and proximity to the ER.

Our literature review showed that the majority of the clustering algorithms deal with static nodes, whereas the clustering algorithms proposed for mobile wireless sensor network have high complexity, thus making them not suitable for the limited computational and energy sensor nodes, needless to mention the excessive delay for making the CH selection process, which is necessary to be minimized, especially in the mobile and dynamic environment of the mobile wireless sensor network.

2.3 Cryptographic Algorithms in WSNs

Providing an efficient cryptographic algorithm that suits the limited WSN node's constraints is challenging, especially that most of the existing and well know algorithms are not well designed and optimized for resource constraints devices. In the literature, several papers [23], [32]–[34] have evaluated the existing algorithms that are based on symmetric, or asymmetric algorithms that use private keys only or private and public keys jointly, and studied their effect on the nodes' energy consumption. In the literature, several research papers have been suggested to provide cryptographic security for small

devices based on the symmetric, public key, or hybrid encryption scheme in restricted environments such as WSNs. Some works [23], [32]–[34] have concentrated on analyzing current encryption algorithms that have a direct effect on WSN nodes' energy and performance. Key management and authentications are other crucial processes that should be optimized to ensure energy efficient yet secure performance of the resources' limited sensor nodes. As such, several researchers proposed efficient key management and authentication mechanisms [35], [36].

Furthermore, several works have been designed with the objective of having lightweight block ciphers with simple and efficient architecture that suits low resources devices. For example,, PRESENT [37], TEA [38], SIT [39], Speck [40] and SIMON [40]. Speck is a lightweight block ciphers algorithm proposed by the National Security Agency (NSA) along with SIMON. Speck is designed to be implemented in software, while SIMON is better implemented via hardware. Speck is an Add–Rotate–XOR (ARX) cipher that supports different blocks and key sizes which both determine the number of rounds. TEA on the other hand uses XOR, ADD, and SHIFT operations to achieve nonlinearity. A block cipher version of TEA known as Block TEA was proposed by Wheeler et al. [41] that can deal with any number of plaintext blocks.

Despite the fact that these lightweight ciphers are specially designed to preserve energy without compromising security. However, some of the proposed lightweight ciphers have suffered attacks on roundest [2] and some of them have shown low performance compared with typical ciphers [32]. Theodore, their adoption in WSN is not without risk. Indeed, a systematic review and evaluation for lightweight ciphers for WSN taking into consideration the ciphers' security and energy efficiency are provided in [42]. The security- performance trade-off for several lightweight ciphers for WSN are studied in [14], [42], [43]. The authors researched the various aspects of lightweight cryptography for limited resource devices in [19] and defined the block, key sizes, number of rounds and the configuration of the cipher as the main parameters affecting ad performance algorithm security. In addition, the authors in [43] assessed the influence of these parameters on the lightweight cipher's architecture and performance. The authors have shown in [42] that a limited block size of 32–64 bits should be selected in a lightweight cipher over the conventional block size of 64–128 bits. In addition, the authors demonstrated that the security of the cipher is largely dependent on the key length. FlexenTech [22] is another lightweight symmetric block cipher considered for resource-limited IoT and WSN devices, where primitive computations are used to encrypt a plaintext.

Shannon [44] proved that the key size should be at least as big as the block size to reach an optimal confidentiality rate. In addition, the authors examined the most powerful key lengths and the lightweight block ciphers in [45]. The authors found that the time needed for the whole key space k to be exhausted is proportional to the time required for 2^k encryption operations to be carried out. Zhang et al. [15] analysed and compared the effects on WSN energy consumption of various cryptographic algorithms. They found that various variables such as packet size, cipher mode of operation, the initialization vector (IV) of the algorithm used, and the efficiency of the communication channel would influence the energy.

Another line of research investigated the effect of various cryptography algorithms on the nodes' and devices' resources usage such as the memory used, processing time and power consumption [46]–[48]. In [46], the energy consumption of AES, DES (Data Encryption Standard), RSA (Rivest-Shamir-Adleman) and RC4 (Rivest Cipher 4) algorithms have been evaluated for WSN. Several key sizes and different cipher block modes and their effects on energy consumption have been investigated. In [47], the authors study the power consumption of deploying several cipher algorithms on Android smartphones where several files with different sizes have been examined. Particularly, the AES finalist algorithm RC6, Twofish, Serpent, and Mars algorithms. The paper results showed that the least power was consumed by the Twofish and RC6 algorithms followed by Mars and Serpent. Another work that compared several ciphering algorithms such as the RSA, RC6, 3-DES and AES was presented in [48]. The comparison metrics were: execution time, memory needed to store the code and data. Their work showed that the RC6 algorithms consumed the least resources. However, the Electronic Code Book (ECB) mode of operation was used which is considered insecure. Another widely used cipher algorithm that showed its effectiveness is the AES developed by the National Institute of Standards and Technology (NIST) [40].

However, none of these lightweight schemes have been analyzed for their effect on WSN network lifetime or WSN node power consumption. In this paper, we will analyze and compare the effect of using Speck128, TEA, AES and FlexenTech along with DynCH algorithm on WSN network lifetime, which both are essential and high relevance, according to our conducted literature review.

3 The Dynch Scheme For Mobile Wireless Sensor Networks

The DynCH framework consists of two key stages: a complex local cluster selection technique is used in the first stage to constantly adjust the CH to match the energy consumption of the nodes and maximize its lifetime. A lightweight symmetric encryption algorithm is used to protect data transfer in the second stage. These symmetric block schemes will be evaluated on the basis of network lifetime complexity and power usage. Moreover, the key exchange process between the WSN nodes, CHs and the Edge router is implemented using a key management protocol [14]. Figure 2 depicts the WSN architecture deploying the DynCH algorithm won its different phases.

As depicted in Fig. 2, the WSN nodes (n_1, n_2,..., n_N), where N is the number of WSN nodes within the ER). The WSN nodes sense the environment and collect relevant data. Then the sensed data are sent to the Edge Router via the CH. The WSN nodes are assumed to be mobile and can move in different directions. In the beginning, the CH is chosen depend on several conditions such as the WSN node energy, the number of neighbouring nodes and the distance between them. Once the CH is chosen and the clusters are formed. The key management exchange process will be used and data are sent securely using one of the lightweight symmetric encryption algorithms.

The first phase of the local cluster selection methodology is illustrated in Fig. 3. The parameters d, d_T, E_T, w, and ω represent the distance between WSN nodes, the distance threshold, the WSN node energy threshold, the average distance of candidate CHs to neighbouring nodes, and the weight of CHs selection,

respectively by the flowchart. However, all these parameters were explained in details in [14]. The algorithm starts by continuous verification (period of time) of the distance between each WSN node (*n*) and its CH.

Since the WSN nodes travel at a low speed of approximately 5 m/s, the distance is tested once per second [49]. If the distance starts reaching the value of the distance threshold (d_T), the WSN node starts searching for the nearest CH to pair with it. To target the CH, a WSN node sends a *Join_Request* message, then sends the *Release_Request* message to its CH source. This process uses WSN mobile nodes that lead to the reassembling and regrouping. If *n* travels away from its CH source and when the distance between them becomes less than d_T, *n* continues to scan for the closest CH, if there is, *n* binds to the CH target and then releases it from the CH source. The target CH is the nearest CH found by *n*. The CH source is a CH node bound by *n*. *n* initiates the process of forming a new CH between the neighboring WSN nodes if there is no nearby CH, and then invites them to join. If the CH moves away from its related WSN nodes, on the other hand, it has a few of them (*c*), where *c* is the lower WSN nodes that are attached to each CH. CH will release the linked WSN nodes and become a periodic WSN node in order to join to the nearest CH. Moreover, the WSN nodes that have been free can also check for or build new cluster classes.

The Low Power and Lossy Networks (RPL) protocol is nevertheless designed to catch devices that merge mesh and tree topologies in 6LoWPAN networks. The root uses the broadcast message DIO in regular RPL to draw an instance of topology [26]. After updating, each WSN node that receives a DIO message will add the sender as its parent and then forward it to its neighboring WSN node [50]. Ultimately, the WSN node chooses the best route to act as the default gateway, depending on the parent list.

This work, however, relies primarily on ER to identify the CHs that are deployed in the ER area from the WSN nodes. This decision will be dispersed to the numerous zoning areas near the ER, after which a self-organized CH rotation will be carried out. Next, by sending the *CH-Announce* message, the CH starts transmitting its identifier within its own province. The distance to the neighboring CHs is measured by neighboring WSN nodes and the closest one is calculated. However, the upper limit of the WSN nodes attached to each CH is not provided by the DynCH scheme.

Since the data in WSN networks is transmitted in unsecured wireless networks, all connections in the networks must be secured. The key management system and encryption schemes are employed should keep the overhead costs of key generation, agreement, distribution and data ciphering in low-power networks. In the second phase of the DynCH scheme, we will use some of the symmetric block ciphers to evaluate their performance on WSN network lifetime along it. Moreover, these cipher schemes are usually containing simple key schedules running on primary processes such as AND or XOR. Even more, they are supporting various block sizes (such as 32, 48, 64, 96, and 128 bit). Therefore, these schemes are often aimed at hardware or software applications. Hardware- associated lightweight ciphers symmetric blocks algorithms include SIMON, LED, Piccolo, and PRESENT, while LEA, Speck, and Chaskey are among the software-associated lightweight ciphers [51]. Most of these encryption schemes are found secure enough to be used in real world applications. The Speck128, TEA, and FlexenTech are used to establish a secure

and reliable connection between WSN nodes and their linked CHs, and ultimately reach the ER. Moreover, the key management algorithm proposed in [14] has been used in this analysis. These symmetric encryption algorithms are selected due [51]to their reliability and lightness compared to others [51].

6LoWPAN uses AES for data encryption. In AES, the block size of this encryption is 128 bits, with key sizes of 128, 192, and 256 bits. The number of rounds depends on key size, 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds when using a 256-bit key. Moreover, it is based on a swap–transformation network structure, whose primary procedures being SubBytes, ShiftRows, MixColumns, and AddRoundKey. The best attack on full AES is the biclique, nonetheless it is slightly better than brute force [23]. In TEA, cryptography emphasis Feistel iteration using 64-bit block size and a 128-bit key size where individual rounds employ K[0; 1] and even rounds employ K[2; 3]. With using 64 rounds (32 cycles), the TEA authors say 16 cycles may be appropriate but they recommend 32 cycles using the constant" C" (to stop simple attacks depend on the symmetry of the rounds) [52]. Moreover, Speck has 10 alternative forms where the block size is 32, 48, 64, 96 or 128 bits, the key size is 64, 72, 96, 128, 144, 192 or 256 bits and the round numbers rely on the given parameters. A block always involves two words and the word size can be 16, 24, 32, 48 or 64 bits. Moreover, the corresponding key consists of 2, 3 or 4 words. Each round involves two cycles, adding the right word to the left word, XORing the key to the left word, then XORing the left word to the right word.

In terms of reconfiguration flexibility, FlexenTech and Speck cryptographic schemes are designed with smoothed flexibility allowing for some different block and key sizes in use. Speck gets its nonlinearity from the standard summation process; key lengths of less than 80 bits have been shown to not offer a high level of security. Hence, the Speck128 is selected for this study. In managing sophistication and level of security, FlexenTech provides stability. The algorithm only conducts simple calculations using a mixture of, substitutes, random permutations and bit-level rotations to encrypt single info [22]. Table 1 shows the key schedule and block-size variations between their techniques.

Table 1
The lightweight symmetric variation keys and block sizes

Ciphers	Block size (bits)	Key Size (bits)	Rounds
Speck128	2*64 = 128	2*64 = 128	32
		3*64 = 192	33
		4*64 = 256	34
TEA	2*32 = 64	128	64
AES	128	128	10
		192	12
		256	14
FlexenTech	16	128	4
	32		8
	128		12
	256		16
	512		32

4 Simulation And Environments

We address the simulation setting in this section and the parameters are used first. The DynCH model is evaluated in order to figure out the lifetime of the WSN network depending on the various simulation conditions. First, the DynCH is constructed, then symmetrical block structures are implemented. The performance of the DynCH scheme is compared to constant clustering techniques [5]–[9] called "SteadyCH".

4.1 Simulation and Performance Metrics

Cooja simulator [53] is used to simulate the wireless network architecture of 6LoWPAN. Cooja is working on a Contiki OS[54] which is an open-source operating system designed for Intent of Thing (IoT) technology to manage and control the device's hardware and software. The simulation runs on a 1.8 GHz Intel Core i7 processing computer with 8 MB of cache and 8 GB of RAM. Table 2 includes the default parameters used in the wireless network architecture, and some of the parameter values in the table are taken from Darabkh et al's values [6].

In the simulation, the different WSN nodes' numbers are initially distributed with dimensions of 500 x 500 terrain connected to different CHs numbers at the initial values as shown in Fig. 4 (a & b). After that, the WSN nodes start moving in a random direction at a rate of 5 meters per second.

In the simulation, the numbers of the different WSN nodes are initially distributed with dimensions of 500 x 500 terrain linked to the initial values of different CHs numbers, as seen in Fig. 4 (a & b). After that, at a rate of 5 m/s, the WSN nodes start traveling in a random direction.

Table 2
Simulation parameters used

Parameter	value
WSN node size	60 m x 120 m
ER location	X = 30, Y = 90
Number of CHs	2, 4, 8, 12, 20
N	100, 200, 300, 400
Simulation time	50, 100, 150, 200, 400
WSN node speed	5 meter/second
Message size	6400 bits
Control message size	200 bits
Initial energy (Joule)	0.5, 0.75, 1, 1.25
Two-ray grounded propagation models	0.0013 PJ/bit/m^4
Free space model	10 PJ/bit/m^2
Power consumed by transmitter	50 nJ/bit
Transition power	20 nJ/bit
Power consumed by receiver	50 nJ/bit
Energy consumption per round	0.001 J
Energy consumption per block size	0.001 for 32 bits
Distance threshold	87 m

4.2 Experimental Results

For key efficiency metrics, the following are used to test the DynCH scheme based on WSN node communications clustering and security and expanding the lifetime of the WSN network.

- Effecting of the number of CHs to network lifetime on both DynCH and SteadyCH mechanisms.
- Evaluate the effecting of the WSN nodes numbers on network lifetime for both DynCH and SteadyCH mechanisms.

- Effecting of different simulation time to network lifetime on both DynCH and SteadyCH mechanisms.
- Execution speed of the lightweight of symmetric encryption schemes on different data sizes.
- Effecting of the lightweight of symmetric encryption schemes to network lifetime on both DynCH and SteadyCH mechanisms.

5 The Results

The lifetime of the network is calculated when the power of some WSN nodes reaches 0. Therefore, in our experimental results, we rely on different parameter values to plot under different conditions. With regard to the evaluation of the “DynCH” proposal scheme, we set the initial energy of the WSN node to 1 joule and the simulation time set to 200 seconds to allow some of the WSN nodes' energy to reach 0. Moreover, the WSN nodes number set to 200 as well.

In the second circumstance, we assess the impact of lightweight security algorithms in relation to the use or without the use of cryptographic algorithms in static clustering "steadyCH" and dynamic clustering "DynCH" techniques. Hence, we set the simulation time to 400 seconds to allow an of WSN nodes energy that does not have an encryption mechanism to reach 0.

5.1 Complexity Analysis for the DynCH on the Network Lifetime

In the following analysis, the DynCH scheme is compared with steadyCH technique in different numbers of CHs, different WSN nodes number, and different execution times. The main objectives of the following results illustrate the effect of DynCH optimization on WSN in different environments.

Initially, in order to examine the effect of different numbers of CH on WSN network lifetime, we set the number of WSN nodes to 200 and run the simulation program on different numbers of CH each time. The specific CH numbers are 2, 4, 8, 12 and 20 as shown in Fig. 5.

As shown in Fig. 5, the network lifetime in the steadyCHs scheme begins to increase from 10.3s when the number of CHs is 2 to 33.7s when the number of CHs becomes 20. This increment in the network lifetime is due the increment of CH numbers and this increment of CHs leads to a distribution of the energy consumption between different CH nodes. Since the CH nodes are responsible for receiving and transmitting data from ordinary nodes to ER, therefore, the double number of CHs increases the network lifetime by 2, as is the case when the number of CHs increase from 4 to 8. Meanwhile, the same Fig. 5 shows that there is no clear change in the network lifetime for the DynCH scheme in different CHs numbers. The network lifetime is 36.9s when the number of CHs is 2 and the network lifetime is 38.1s when the number of CHs is 20. This change in network lifetime differentiation does not occur due to the dynamic selection of appropriate CHs while the simulation time is 200s. When the WSN nodes move in 5 m/s, the nodes start to move away from thier CH. The DynCH scheme advises each node to rejoin to

another closest CH or selecting a new CH among adjacent WSN nodes, whereas in the steadyCHs mechanism, each node remains connected to its CH and the distance increases will increase the consuming WSN node energy. The result in Fig. 5 shows the DynCH improves network lifetime by 72%, 66%, 52%, 44%, and 12% compared to SteadyCHs in different numbers of CHs, respectively.

The analysis effect of WSN nodes number to the network lifetime is depicted in Fig. 6. The number of nodes is increased by 100 nodes each time and the number of CHs is set to 8 CHs for both schemes. As illustrated from Fig. 6, an increase in WSN nodes affects both techniques by reducing the network lifetime. The reason behind this decrement is the effect of over-connecting the ordinary WSN nodes to the CHs. However, the DynCH algorithm shows the improvement of network lifetime compared to the steadyCH technique because of the CH dynamic selection. Nevertheless, Fig. 6 shows a decrement in network lifetime as the WSN nodes increase from 200 to 300. The reasons behind this are the density of distributed WSN nodes and the maximum number of WSN nodes parameter is not taken into account for each CH node. Each CH can accept any WSN node whose d is less than the threshold value.

Finally, the result shows the DynCH improves network lifetime by 46%, 51%, 40%, and 49% compared to SteadyCHs in a different number of WSN nodes (100, 200,300, and 400), respectively.

In order to allow the WSN nodes' energy to reach 0 when the simulation time is less than 200 seconds for network lifetime detection, we need to reduce the WSN nodes' initial energy. Therefore, in Fig. 7, the WSN nodes' initial energy is set at 0.5 J, the CHs number is set at 8, and the WSN nodes' number is set at 200 as well. In addition, the simulation program runs a different number of simulation times each time. The specific simulation times are 50, 100, 150, and 200 seconds.

As illustrated from the same Fig. 7, the effect of simulation time on the network lifetime in both schemes is slightly different. The network lifetime in DynCH begins to increase when the simulation time increases while in SteadyCHs starts to decrease. This differentiation is due to the positive relationship between the DynCH function and the duration time. The duration time will increase the distance (d) between WSN nodes and their CHs, thus, the DynCH function will be launched and then start to create new CHs or join other CHs. This process will save power in both nodes (CH and ordinary), while in steadyCH the increment of duration time will increase the distance between WSN nodes and their CHs, thus, the consuming power will increase and the network lifetime will decrease.

The result shows the DynCH and steadyCH started slightly the same when the simulation time at the 50s. The DynCH then starts to improve the network lifetime by 22%, 31%, and 53% compared to steadyCH in a different number of WSN nodes, respectively.

5.2 Complexity Analysis for the lightweight encryption algorithms on the DynCH scheme for extending lifetime

We test the performance efficiency of the Speck128, FlexenTech, TEA, and regular AES encryption algorithms relative to each other during the encryption and decryption time. With rounds ranging from 4

to 32 and block sizes ranging from 4 to 128, the average measurement times produced by encrypting data of different sizes will be taken into account. Since applying the Speck128, FlexenTech, TEA and AES systems, the findings are linked with the encryption times obtained. For each of the specified key sizes, the AES utilizes a 128-bit block size and customizable round numbers. With key lengths of 128 bits, 192 bits and 256 bits respectively, it uses 10, 12, and 14 rounds. The TEA uses a 128-bit key and a block size of 64 bits in 64 rounds a Feistel structure type. Meanwhile, the rounds' number in Speck128 depends on both block and key size, and in FlexenTech, any number of rounds or key sizes can be used. Figure 8 shows the comparison of the encryption times between Speck128, FlexenTech cipher, TEA, and AES encryption algorithms.

From the results, we have noticed that the Speck128 outperforms FlexenTech, TEA and AES. In the encryption method, both TEA and AES use constant parameters, while variations based on key and block sizes are used by FlexenTech and Speck128. The Speck128 can achieve the average encryption throughput of 121.1 byte/ms compared to 91.23 byte/ms, 62.36 byte/ms and 40.78 byte/ms for the FlexenTech, TEA, and AES ciphers respectively.

In the following circumstance, we analyze the effect of lightweight symmetric encryption on the DynCH and SteadyCH mechanisms. The number of CHs was set at 8 and the rest of the simulation parameters were discussed above in Sect. 5.0. Figure 9 explains the difference in network lifetime using Non-encryption (N/A), Speck128, FlexenTech, TEA, and AES on SteadyCH and DynCH algorithms in different initial power (0.5 J to 1.25 J).

Based on the same Fig. 9, the increase in the initial power increases the network lifetime in all cases and both techniques (SteadyCH and DynCH). The highest network lifetime is N/A, as there is no power to lose for encryption and decryption processes. Meanwhile the symmetric encryption algorithms increase the network consuming power invariant values based on their techniques. Therefore, Figures in (9.a) and (9.b) show slightly the same variation between encryption algorithms in network lifetime. Meanwhile, the difference in network lifetime between SteadCH and DynCH belongs to DynCH features which improve communication between nodes in the 6LoWPAN protocol. DynCH improves the network lifetime by 45% compare to SteadyCH in different initial power.

Regarding the lightweight encryption algorithm, the best algorithm that saves power and has the highest network lifetime is Speck128. The Speck128 has the lowest execution time in different block symmetric ciphers in all different data sizes. The Speck128 improves the average network lifetime by 34%, 52%, and 70% compared to FlexenTech, TEA, and AES, respectively. Moreover, the FlexenTech improves the network lifetime by 27% and 0.54% compared to TEA and AES respectively. The TEA improves the network lifetime by 37% compared to AES.

In addition, regarding symmetric block algorithms power consumption, Speck128 uses 26%, FlexenTech uses 52%, TEA used 65%, and AES used 78% compared to N/A, respectively.

6 Conclusion And Future Work

In this paper, we have analyzed the DynCH scheme in WSN networks' lifetime and saving energy in the 6LoWPAN protocol. The DynCH depends on distances and nodes' power to reformulate CH nodes between WSN nodes, which keep the WSN nodes' connection and their energy close to the best value. Moreover, the analyses of lightweight symmetric block algorithms were done to evaluate their power consumption in wireless sensor networks. The lightweight key management was used to unify the authentication method in different symmetric block algorithms and reduce power consumption in establishing and managing cryptographic key between WSN nodes. The Cooja simulator have used to obtain the network lifetime and symmetric encryption time in different environments. The results of DynCH analyses show that improvement of network lifetime by 58% on an average of using different CHs, 47% on an average of using different numbers of WSN nodes, 36% on an average of using different simulation times, and 45% on an average of using different initial power compared to SeadyCH approach. Moreover, the Speck128 algorithm shows the best results in execution time and network lifetime prolonging. The Speck128 improved the average network lifetime by 34%, 52%, and 70% compared to FlexenTech, TEA, and AES, respectively. In addition, the Speck128 used 26%, FlexenTech used 52%, TEA used 65%, and AES used 78% of the WSN network's average lifetime compared to Non-encryption connection, respectively.

In future research, we intend to update DynCH to cover the minimum and the maximum number of WSN nodes in each CH. Moreover, we intend to update it to cover different ERs with alteration of handover process to be suitable with wireless senor network. The Software Defined Network (SDN) controller could be used for this proposal.

References

1. Yi, L., Tong, X., Wang, Z., Zhang, M., Zhu, H., & Liu, J. (2019). A novel block encryption algorithm based on chaotic S-Box for wireless sensor network. *IEEE access : practical innovations, open solutions, 7*, 53079–53090. doi:10.1109/ACCESS.2019.2911395.
2. Lee, C. C. (2020). Security and privacy in wireless sensor networks: Advances and challenges. *Sensors (Basel, Switzerland), 20*(3), 10–12. doi:10.3390/s20030744.
3. Al-Kashoash, H. A. A., Kharrufa, H., Al-Nidawi, Y., & Kemp, A. H. (2019). Congestion control in wireless sensor and 6LoWPAN networks: toward the Internet of Things. *Wirel. Networks, 25*(8), 4493–4522. doi:10.1007/s11276-018-1743-y.
4. Bouaziz, M., & Rachedi, A. (2016). A survey on mobility management protocols in Wireless Sensor Networks based on 6LoWPAN technology. *Computer Communications, 74*, 3–15. doi:10.1016/j.comcom.2014.10.004.
5. Sujanthi, S., & Nithya Kalyani, S. SecDL: QoS-Aware Secure Deep Learning Approach for Dynamic Cluster-Based Routing in WSN Assisted IoT, Vol. 114, 3. Springer US, 2020.

6. Darabkh, K. A., El-Yabroudi, M. Z., & El-Mousa, A. H. (2019). *BPA-CRP: A balanced power-aware clustering and routing protocol for wireless sensor networks* (Vol. 82). Elsevier B.V.
7. Darabkh, K. A., Al-Maaitah, N. J., Jafar, I. F., & Khalifeh, A. F. (2018). EA-CRP: A Novel Energy-aware Clustering and Routing Protocol in Wireless Sensor Networks. *Computers & Electrical Engineering, 72*, 702–718. doi:10.1016/j.compeleceng.2017.11.017.
8. Jain, T. K., Saini, D. S., & Bhooshan, S. V., "Cluster head selection in a homogeneous wireless sensor network ensuring full connectivity with minimum isolated nodes," *J. Sensors*, vol. 2014, 2014, doi: 10.1155/2014/724219.
9. Darabkh, K. A., Albtoush, W. Y., & Jafar, I. F. (2017). Improved clustering algorithms for target tracking in wireless sensor networks. *J. Supercomput., 73*(5), 1952–1977. doi:10.1007/s11227-016-1898-1.
10. Abuarqoub, A., Hammoudeh, M., Adebisi, B., Jabbar, S., Bounceur, A., & Al-Bashar, H. (2017). Dynamic clustering and management of mobile wireless sensor networks. *Comput. Networks, 117*, 62–75. doi:10.1016/j.comnet.2017.02.001.
11. Zhao, Y., Liu, K., Xu, X., Yang, H., & Huang, L. "Distributed dynamic cluster-head selection and clustering for massive IoT access in 5G networks," Appl. Sci., 9, 1, 2019, doi:10.3390/app9010132.
12. Elhoseny, M., Farouk, A., Zhou, N., Wang, M. M., Abdalla, S., & Batle, J. (2017). Dynamic Multi-hop Clustering in a Wireless Sensor Network: Performance Improvement. *Wirel. Pers. Commun., 95*(4), 3733–3753. doi:10.1007/s11277-017-4023-8.
13. Bozorgi, S. M., Shokouhi Rostami, A., Hosseinabadi, A. A. R., & Balas, V. E. (2017). A new clustering protocol for energy harvesting-wireless sensor networks. *Computers & Electrical Engineering, 64*, 233–247. doi:10.1016/j.compeleceng.2017.08.022.
14. Khashan, O. A., Ahmad, R., & Khafajah, N. M. (Apr. 2021). An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks. *Ad Hoc Networks, 115*, 102448. doi:10.1016/j.adhoc.2021.102448.
15. Zhang, X., Heys, H. M., & Li, C. (2012). Energy efficiency of encryption schemes applied to wireless sensor networks. *Secur. Commun. Networks, 5*(7), 789–808. doi:10.1002/sec.375.
16. Khashan, O. A., Zin, A. M., & Sundararajan, E. A. (2014). Performance study of selective encryption in comparison to full encryption for still visual images. *J. Zhejiang Univ. Sci. C, 15*(6), 435–444. doi:10.1631/jzus.C1300262.
17. Yu, D., Kang, J., & Dong, J., "Service Attack Improvement in Wireless Sensor Network Based on Machine Learning," *Microprocess. Microsyst.*, vol. 80, no. December 2020, 2021, doi: 10.1016/j.micpro.2020.103637.
18. Premkumar, M., & Sundararajan, T. V. P. "DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks," Microprocessors and Microsystems, 79, no. August, 2020, doi:10.1016/j.micpro.2020.103278.
19. Dhanda, S. S., Singh, B., & Jindal, P. Lightweight Cryptography: A Solution to Secure IoT, Vol. 112, 3. Springer US, 2020.

20. Gao, M., & Feng, D. (2018). Stochastic stability analysis of networked control systems with random cryptographic protection under random zero-measurement attacks. *Front. Inf. Technol. Electron. Eng., 19*(9), 1098–1111. doi:10.1631/FITEE.1700334.
21. Sun, D.-Z., & Mu, Y. (2020). On the Security of Symmetric Encryption Against Mass Surveillance. *IEEE access : practical innovations, open solutions, 8*, 175625–175636. doi:10.1109/access.2020.3025848.
22. Medileh, S., et al. (2020). A flexible encryption technique for the internet of things environment. *Ad Hoc Networks, 106*, 102240. doi:10.1016/j.adhoc.2020.102240.
23. Saraiva, D. A. F., Leithardt, V. R. Q., de Paula, D., Mendes, A. S., González, G. V., & Crocker, P., "PRISEC: Comparison of symmetric key algorithms for IoT devices," *Sensors (Switzerland)*, vol. 19, no. 19, pp. 1–23, 2019, doi: 10.3390/s19194312.
24. Ashibani, Y., & Mahmoud, Q. H. (2017). Cyber physical systems security: Analysis, challenges and solutions. *Comput. Secur., 68*, 81–97. doi:10.1016/j.cose.2017.04.005.
25. Kumar, V., & Tiwari, S., "Routing in IPv6 over low-power wireless personal area networks (6LoWPAN): A survey," *J. Comput. Networks Commun.*, vol. 2012, 2012, doi: 10.1155/2012/316839.
26. Olsson, J., "6LoWPAN demystified," 2014. [Online]. Available: http://www.ti.com/lit/wp/swry013/swry013.pdf.
27. Kumar, P. M., & Gandhi, U. D. (2020). Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application. *J. Supercomput., 76*(6), 3963–3983. doi:10.1007/s11227-017-2169-5.
28. Mazumdar, N., & Om, H. "DUCR: Distributed unequal cluster-based routing algorithm for heterogeneous wireless sensor networks," Int. J. Commun. Syst., 30, 18, 2017, doi:10.1002/dac.3374.
29. Yarinezhad, R., & Hashemi, S. N. (2019). A routing algorithm for wireless sensor networks based on clustering and an fpt-approximation algorithm. *Journal of Systems and Software, 155*, 145–161. doi:10.1016/j.jss.2019.05.032.
30. Priyadarshi, R., Rawat, P., & Nath, V. (2019). Energy dependent cluster formation in heterogeneous wireless sensor network. *Microsystem Technologies, 25*(6), 2313–2321. doi:10.1007/s00542-018-4116-7.
31. Arumugam, G. S., & Ponnuchamy, T., "EE-LEACH: development of energy-efficient LEACH Protocol for data gathering in WSN," *Eurasip J. Wirel. Commun. Netw.*, vol. 2015, no. 1, pp. 1–9, 2015, doi: 10.1186/s13638-015-0306-5.
32. Biswas, K., Muthukkumarasamy, V., Wu, X. W., & Singh, K. (2016). Performance evaluation of block ciphers for wireless sensor networks. *Adv. Intell. Syst. Comput., 452*, 443–452. doi:10.1007/978-981-10-1023-1_44.
33. Aboshosha, B. W., Dessouky, M. M., & Elsayed, A. (2019). Energy Efficient Encryption Algorithm for Low Resources Devices. *Acad. Res. Community Publ., 3*(3), 26. doi:10.21625/archive.v3i3.520.
34. Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized*

Computing, 0(0), 1–18. doi:10.1007/s12652-017-0494-4.

35. Doerr, L., Heigl, M., Fiala, D., & Schramm, M., "Comparison of energy-efficient key management protocols for wireless sensor networks," *ACM Int. Conf. Proceeding Ser.*, pp. 21–26, 2019, doi: 10.1145/3343147.3343156.
36. Messai, M. L., & Seba, H. (2016). A survey of key management schemes in multi-phase wireless sensor networks. *Comput. Networks, 105*, 60–74. doi:10.1016/j.comnet.2016.05.005.
37. Bogdanov, A., et al. (2007). "PRESENT: An Ultra-Lightweight Block Cipher. In *" in Cryptographic Hardware and Embedded Systems - CHES 2007* (pp. 450–466). Berlin: Springer Berlin Heidelberg.
38. Wheeler, D. J., & Needham, R. M. (1995). Tea, a tiny encryption algorithm. *Lecture Notes in Computer Science, 1008*, 363–366. doi:10.1007/3-540-60590-8_29.
39. Usman, M., Ahmed, I., Imran Aslam, M., Khan, S., & Shah, U. A. (2017). SIT: A lightweight encryption algorithm for secure internet of things. *arXiv, 8*(1), 1–10. doi:10.14569/ijacsa.2017.080151.
40. Bogdanov, A., Khovratovich, D., & Rechberger, C. (2011). Biclique cryptanalysis of the full AES. *Lecture Notes in Computer Science, 7073 LNCS*, 344–371. doi:10.1007/978-3-642-25385-0_19.
41. Song, L., Huang, Z., & Yang, Q. (2016). Automatic differential analysis of ARX block ciphers with application to SPECK and LEA. *Lecture Notes in Computer Science, 9723*, 379–394. doi:10.1007/978-3-319-40367-0_24.
42. Cazorla, M., Marquet, K., & Minier, M., "Survey and benchmark of lightweight block ciphers for wireless sensor networks," *ICETE* 2013 *– 10th Int. Jt. Conf. E-bus. Telecommun. SECRYPT 2013 – 10th Int. Conf. Secur. Cryptogr. Proc.*, vol. 2, no. 1, pp. 543–548, 2013, doi: 10.5220/0004530905430548.
43. Pei, C., Xiao, Y., Liang, W., & Han, X., "Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks," *Eurasip J. Wirel. Commun. Netw.*, vol. 2018, no. 1, 2018, doi: 10.1186/s13638-018-1121-6.
44. Shannon, C. E. (1949). Communication Theory of Secrecy Systems. *Bell System Technical Journal, 28*(4), 656–715. doi:10.1002/j.1538-7305.1949.tb00928.x.
45. Huang, J. L., & Lai, X. J. (2014). What is the effective key length for a block cipher: An attack on every practical block cipher. *Sci. China Inf. Sci., 57*(7), 1–11. doi:10.1007/s11432-014-5096-6.
46. Sibahee, M. A. A., Lu, S., Hussien, Z. A., Hussain, M. A., Mutlaq, K. A. A., & Abduljabbar, Z. A., "The best performance evaluation of encryption algorithms to reduce power consumption in WSN," *Proc. – 2017 Int. Conf. Comput. Intell. Inf. Syst. CIIS* 2017, vol. 2018-Janua, pp. 308–312, 2018, doi: 10.1109/CIIS.2017.50.
47. Soewito, B., Gunawan, F. E., Diana, & Antonyova, A., "Power consumption for security on mobile devices," *Proc. – 11th* 2016 *Int. Conf. Knowledge, Inf. Creat. Support Syst. KICSS 2016*, pp. 4–7, 2017, doi: 10.1109/KICSS.2016.7951435.
48. Ochoa, I. S., Leithardt, V. R. Q., Zeferino, C. A., & Silva, J. S., "Data transmission performance analysis with smart grid protocol and cryptography algorithms," 2018 *13th IEEE Int. Conf. Ind. Appl. INDUSCON 2018 - Proc.*, pp. 482–486, 2019, doi: 10.1109/INDUSCON.2018.8627195.

49. Ahmad, R., Sundararajan, E. A., Othman, N. E., & Ismail, M. “An efficient handover decision in heterogeneous LTE-A networks under the assistance of users’ profile,” Telecommun. Syst., 68, 1, 2018, doi:10.1007/s11235-017-0374-4.
50. Mayzaud, A., Badonnel, R., & Chrisment, I. (2016). A taxonomy of attacks in RPL-based internet of things. *Int. J. Netw. Secur., 18*(3), 459–473.
51. Dinu, D., Le Corre, Y., Khovratovich, D., Perrin, L., Großschädl, J., & Biryukov, A. (2019). Triathlon of lightweight block ciphers for the Internet of things. *J. Cryptogr. Eng., 9*(3), 283–302. doi:10.1007/s13389-018-0193-x.
52. Hong, S., Hong, D., Ko, Y., Chang, D., Lee, W., & Lee, S. (2004). Differential cryptanalysis of tea and XTEA. *Lecture Notes in Computer Science, 2971*, 402–417. doi:10.1007/978-3-540-24691-6_30.
53. “https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja.”.
54. Bin Zikria, Y., Afzal, M. K., Ishmanov, F., Kim, S. W., & Yu, H. (2018). A survey on routing protocols supported by the Contiki Internet of things operating system. *Futur. Gener. Comput. Syst., 82*, 200–219. doi:10.1016/j.future.2017.12.045.

Figures

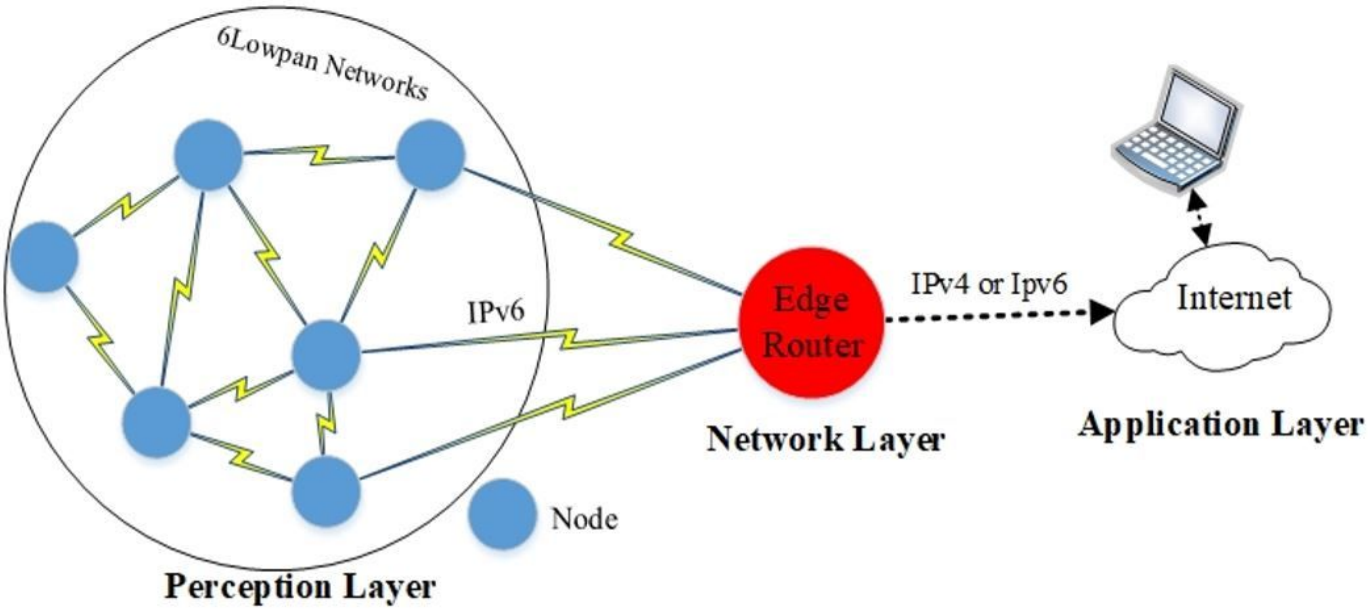


Figure 1

WSN running the 6LoWPAN Protocol

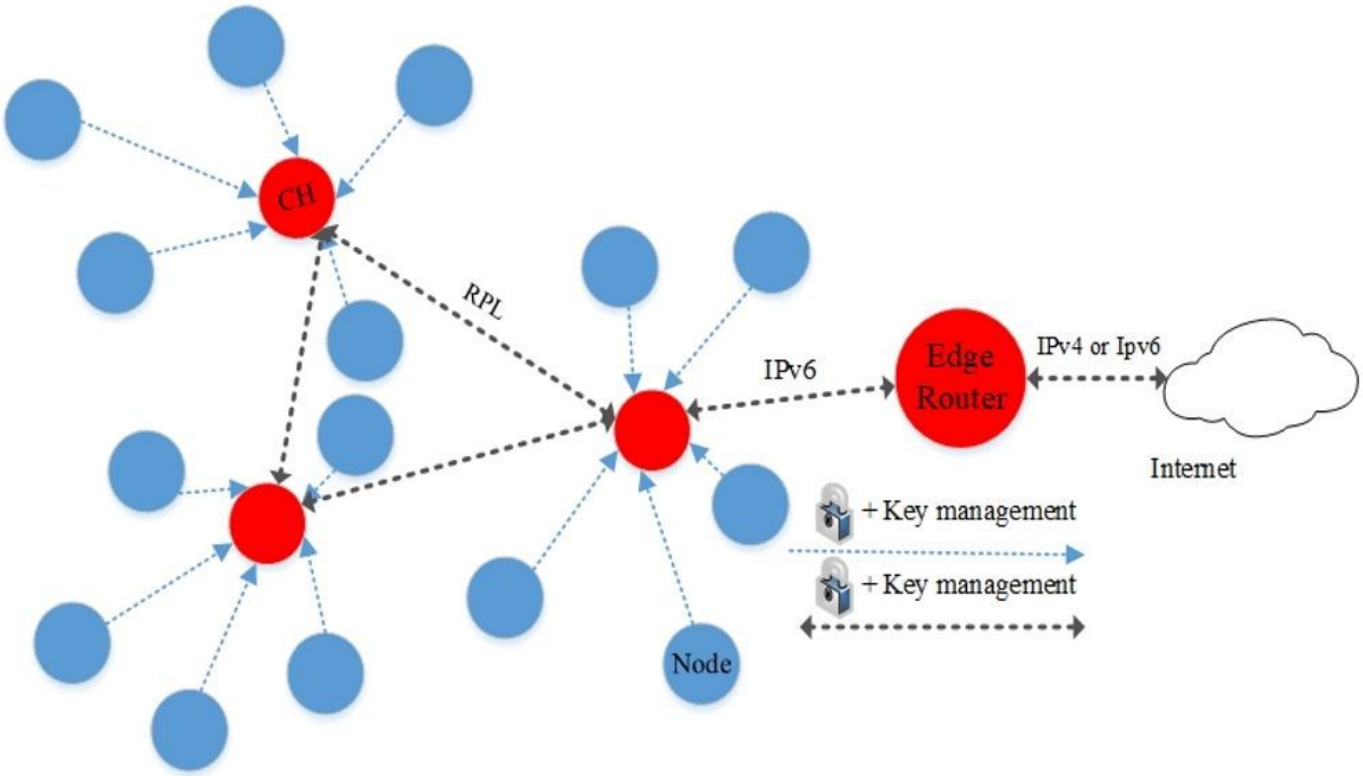


Figure 2

The WSN architecture deploying DynCH algorithm

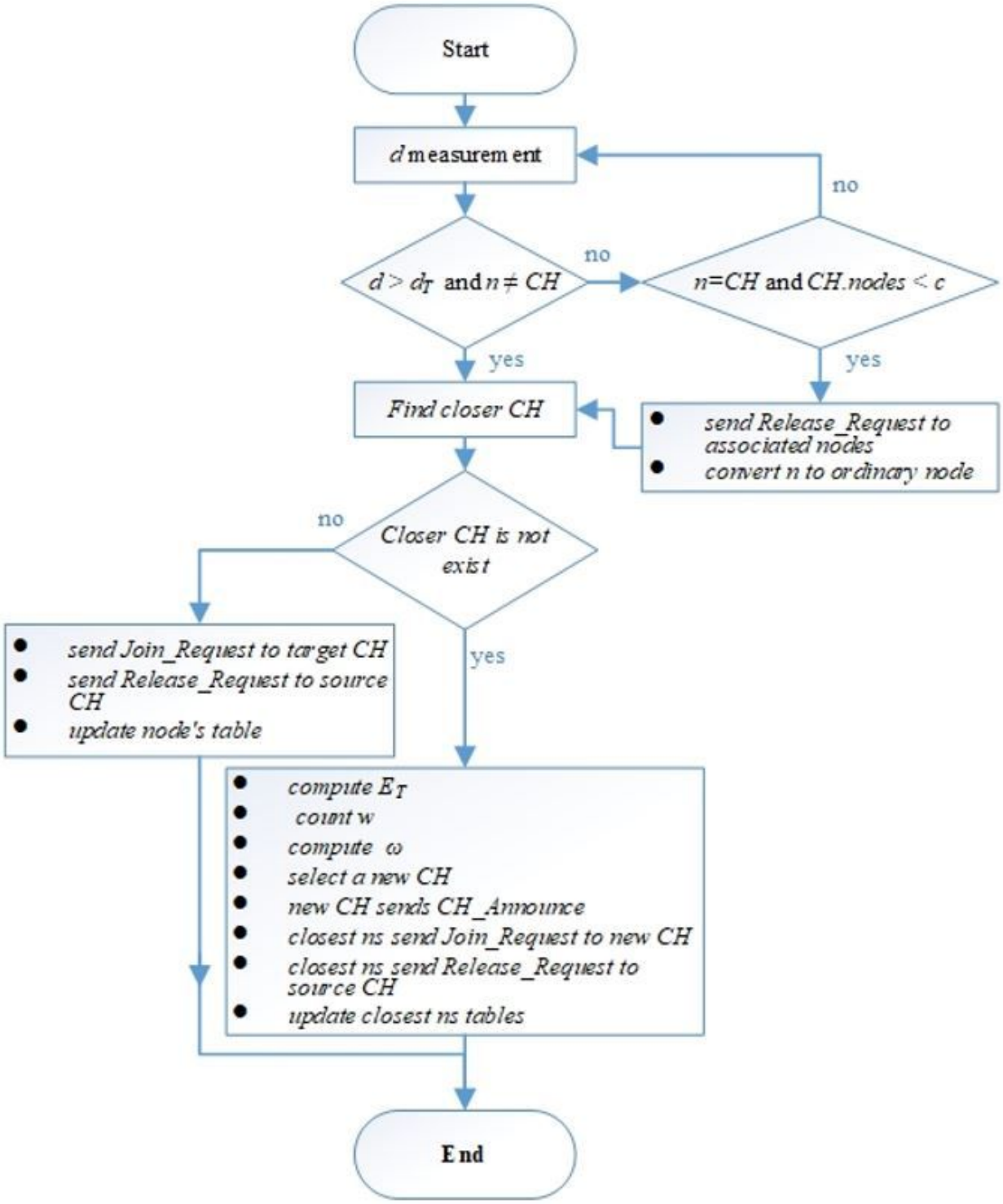
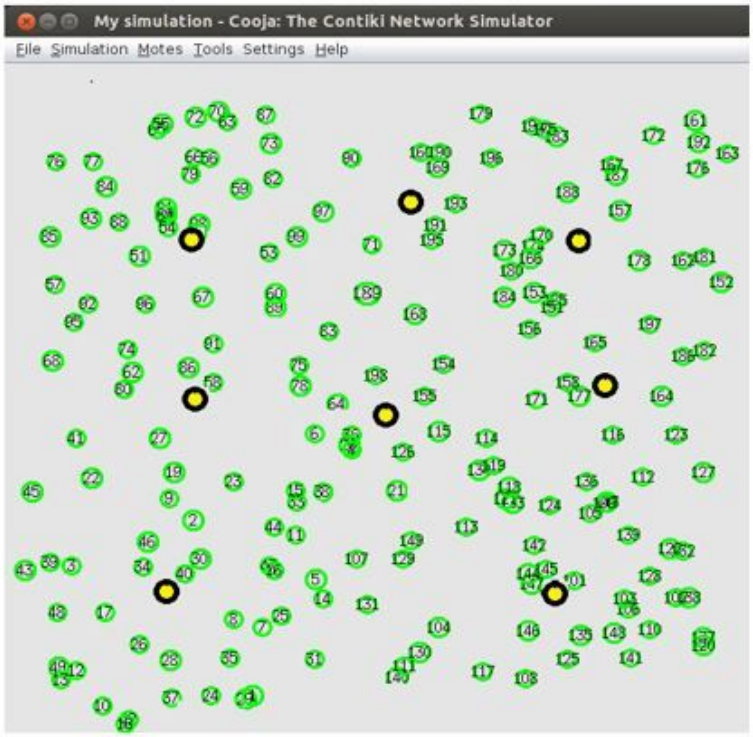


Figure 3

the DynCH scheme

(a)



(b)

Figure 4

Simulation WSN nodes in different CHs and WSN nodes numbers a) 300 WSN nodes with 12 CHs b) 200 WSN nodes with 8 CHs

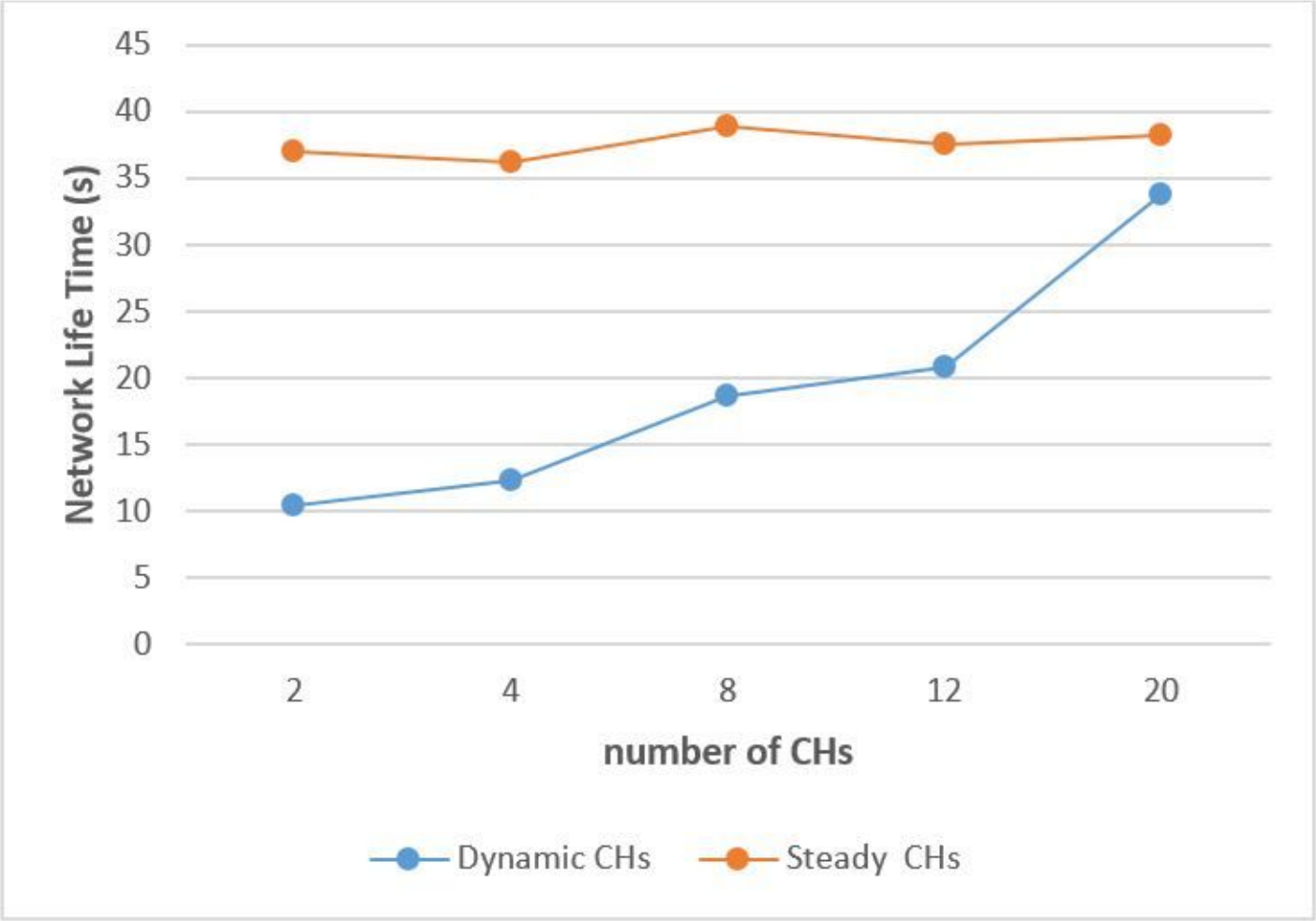


Figure 5

Analysis of the effect of number of CHs on the network lifetime

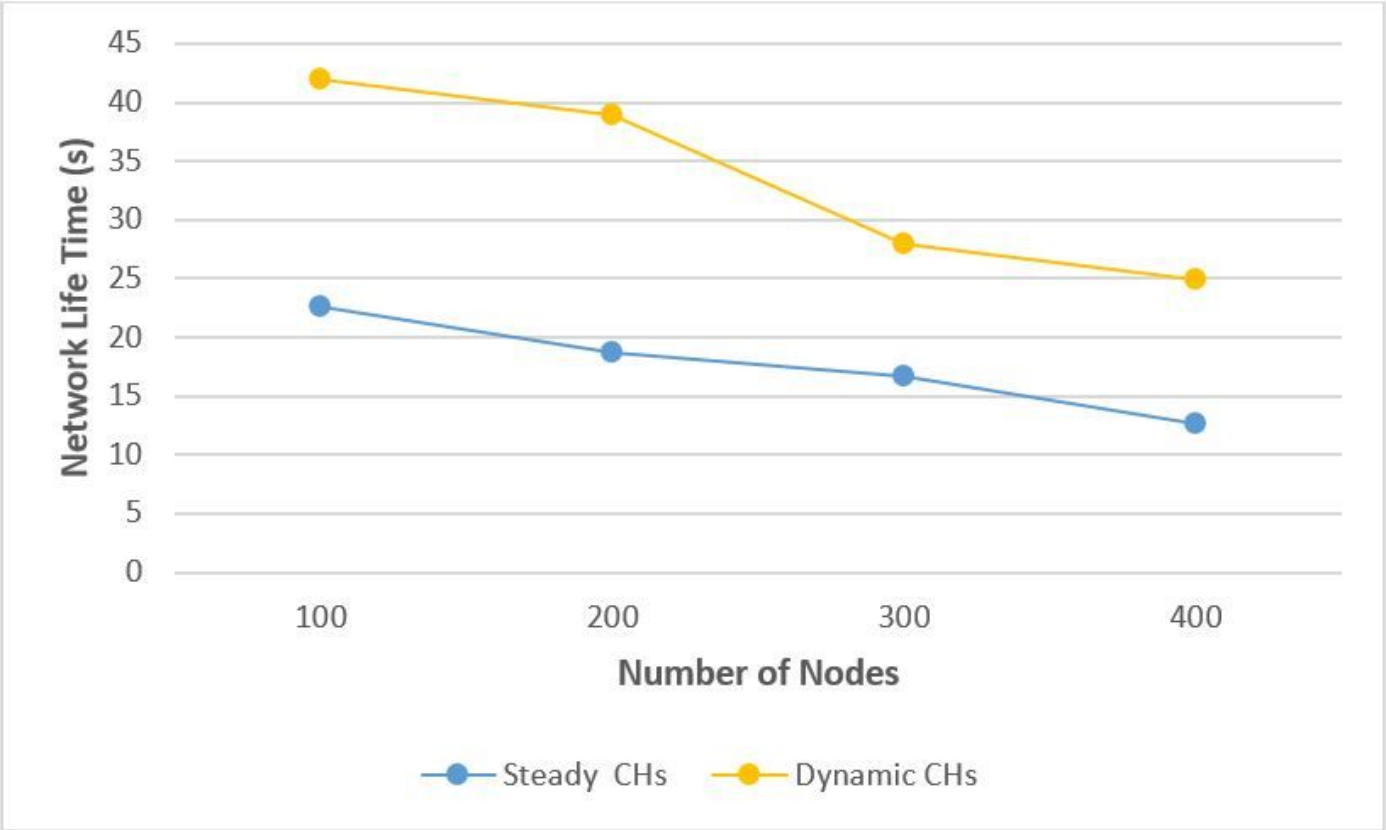


Figure 6

Analysis of the effect of WSN nodes number on the network lifetime

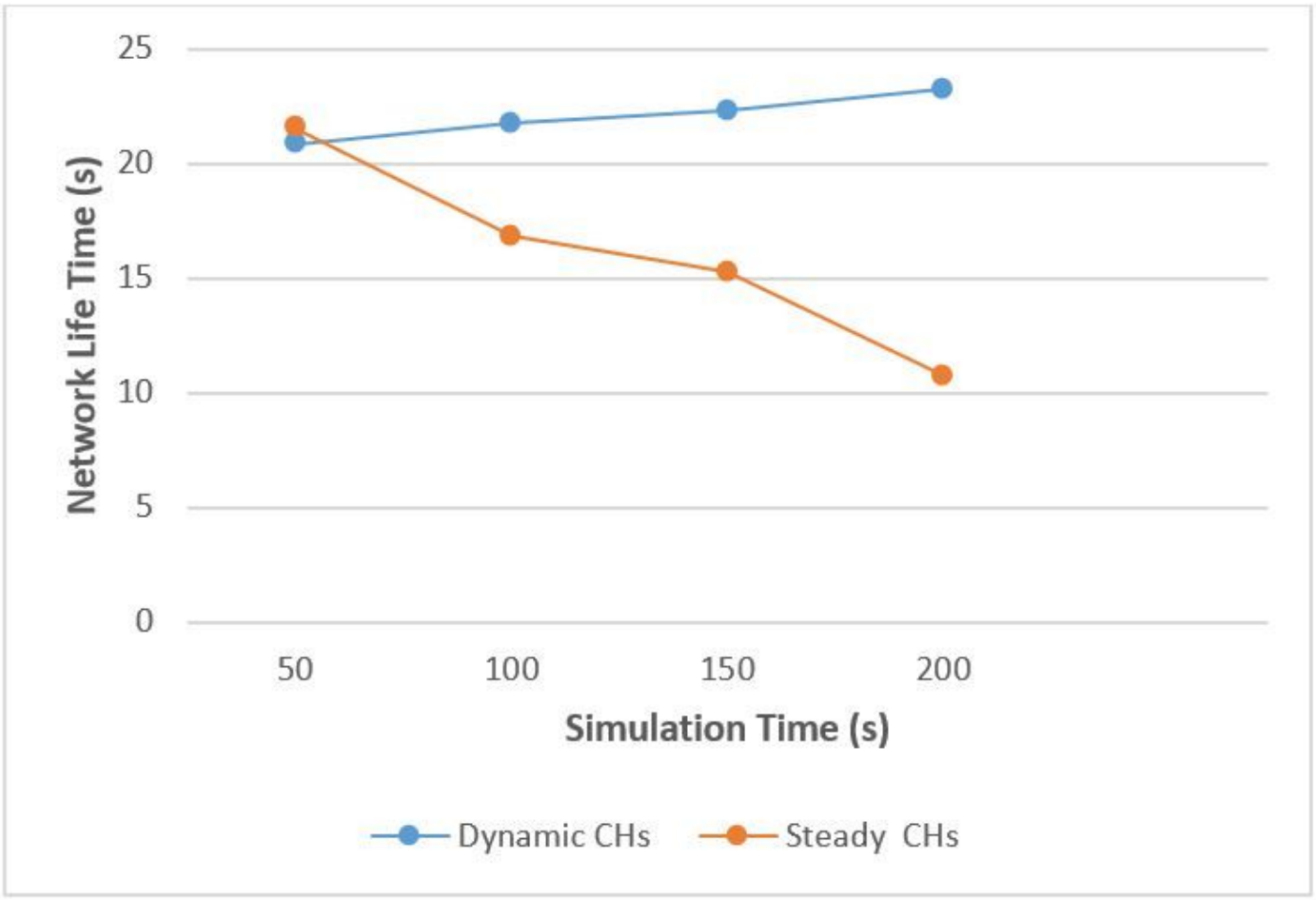


Figure 7

Analysis of the effect of simulation time on the network lifetime

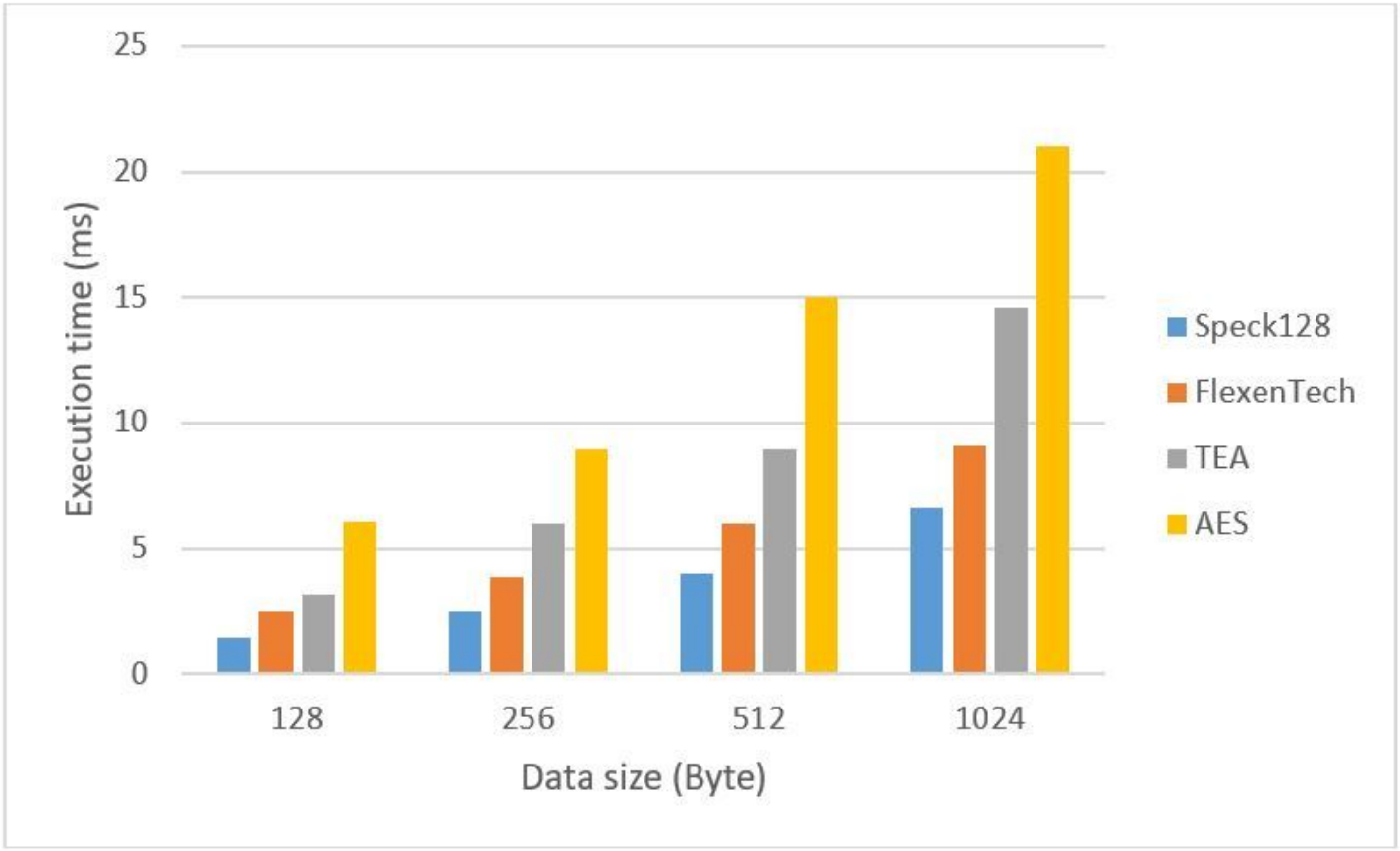
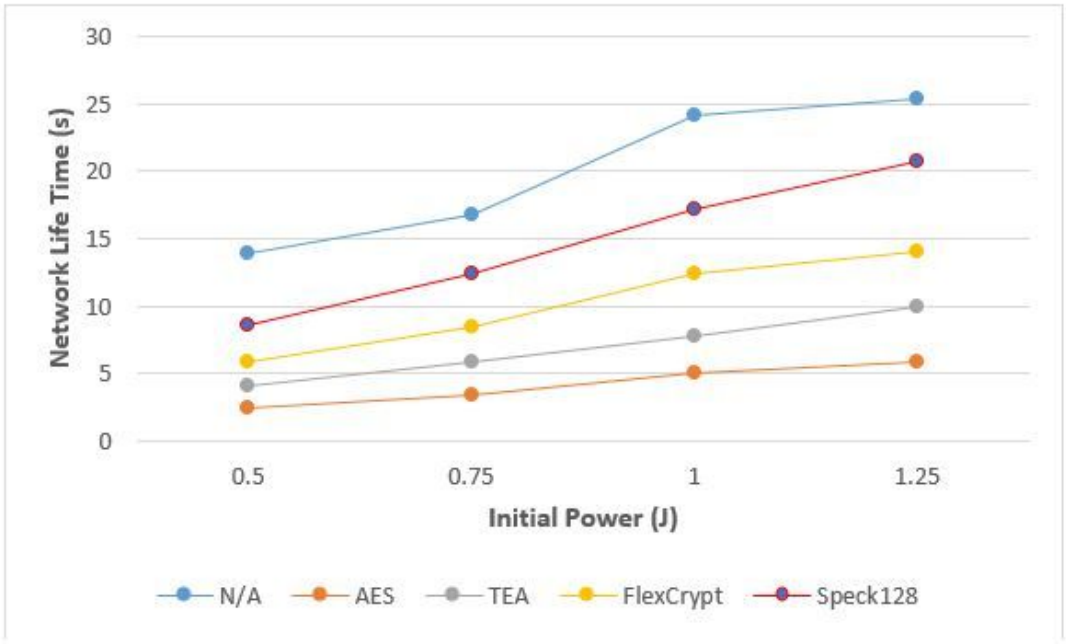
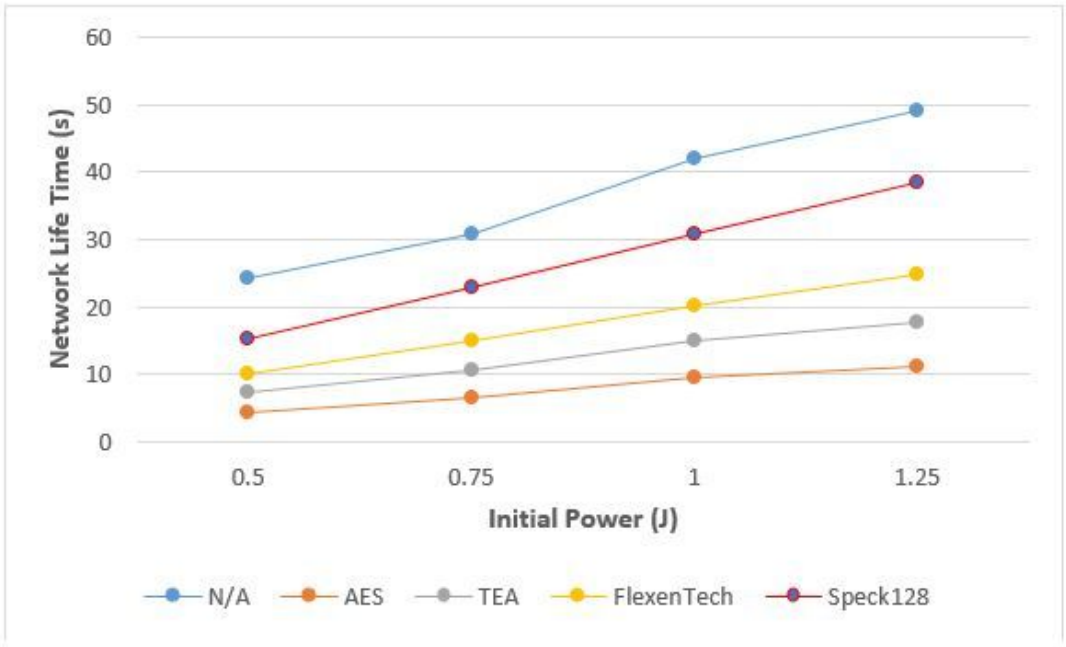


Figure 8

Comparison of encryption times in milliseconds using data of various sizes using Speck128, FlexenTech, TEA and AES encryption algorithms.



(a)



(b)

Figure 9

Analysis the effect of different lightweight symmetric algorithms on network lifetime in a) SteadyCH and b) DynCH schemes.