electronics

MDPI

Article

A Perfect Security Key Management Method for Hierarchical Wireless Sensor Networks in Medical Environments

Raad A. Muhajjar [1], Nahla A. Flayh [2] and Mishall Al-Zubaidie [3,*]

[1] Department of Computer Science, Faculty of Computer Science and Information Technology, University of Basrah, Basrah 61004, Iraq
[2] Department of Computer Information System, Faculty of Computer Science and Information Technology, University of Basrah, Basrah 61004, Iraq
[3] Department of Computer Sciences, Education College for Pure Sciences, University of Thi-Qar, Nasiriyah 64001, Iraq
* Correspondence: mishall_zubaidie@utq.edu.iq; Tel.: +964-61-469-869-029

Abstract: Wireless sensor networks (WSNs) have developed during the past twenty years as a result of the accessibility of inexpensive, short-range, and simple-to-deploy sensors. A WSN technology sends the real-time sense information of a specific monitoring environment to a backend for processing and analysis. Security and management concerns have become hot topics with WSN systems due to the popularity of wireless communication channels. A large number of sensors are dispersed in an unmonitored medical environment, making them not safe from different risks, even though the information conveyed is vital, such as health data. Due to the sensor's still limited resources, protecting information in WSN is a significant difficulty. This paper presents a hierarchical key management method for safeguarding heterogeneous WSNs on hybrid energy-efficient distributed (HEED) routing. In the proposed method, the Bloom scheme is used for key management and a pseudo-random number generator (PRNG) to generate keys in an efficient method to keep sensor resources. In addition, using cipher block chaining-Rivest cipher 5 (CBC-RC5) in this method achieved cryptography goals such as confidentiality. A comparison is made between the proposed and existing methods such as dynamic secret key management (DSKM) and smart security implementation (SSI) under the same circumstance to determine the performance of the new method. The data transmission in WSN consumes about 71 percent of a sensor's energy, while encryption computation consumes only 2 percent. As a result, our method reduces the frequency with which data transmissions are made during the key management process. The simulation findings demonstrated that, in comparison to earlier techniques, the proposed method is significantly more secure, flexible, scalable, and energy-efficient. Our proposed method is also able to prevent classifications of node capture attacks.

Keywords: bloom; cipher-block chaining (CBC); HEED protocol; heterogeneous WSN; key management; PRNG; rivest-cipher5 (RC5); WSNs

check for updates

Citation: Muhajjar, R.A.; Flayh, N.A.; Al-Zubaidie, M. A Perfect Security Key Management Method for Hierarchical Wireless Sensor Networks in Medical Environments. *Electronics* **2023**, *12*, 1011. https://doi.org/10.3390/electronics12041011

Academic Editors: Tomasz Rak and Dariusz Rzońca

Received: 29 January 2023
Revised: 9 February 2023
Accepted: 15 February 2023
Published: 17 February 2023

1. Introduction

Wireless sensor networks have received much interest for using them in a diversity of environments, including health, military, agriculture monitoring, and industrial. However, health systems are making use of WSN technology significantly in recent years. In instances where these sensors are able to communicate with one another wirelessly, the sensed health data from the surrounding area is sent to the sink to be treated either through a single hop if the sink is within the same node's range or through multiple hops if the sink is outside of that range [1,2]. The data routing protocol in the WSN is also important to reduce the load on sensor sources [3]. Because of the medium, any hacker can break through the network and obtain its health data [4]. Passive (eavesdropping hackers on the health data exchanged between the parties without alteration or injecting false data into the network) and active are two kinds of attacks (eavesdropping hackers on the health data and conducting malicious events with it before retransmitting to nodes for the purpose of network destruction or grabbing them) [5,6]. In order to transmit important data in this kind of network, it should keep the information as possible, as well as prevent unwanted parties from providing fake information to sensors. Cryptography technical is used to address some of the health network's security challenges, providing confidence [7] and authentication. The sensors have limited resources from where the processing, energy, and memory [8]. Therefore, considered cryptography traditional techniques are not suitable in WSN because they need additional connection, memory, and processing. As a result, when designing and implementing a key management method, it is important to keep in mind the restricted resources available on these devices [9–12].

Cryptographic mechanisms require efficient key management. Secure communications may be compromised by inadequate key management, resulting in key disclosure to attackers. WSN communication threats and vulnerabilities can be mitigated through key management, which is an essential process. Confidentiality, integrity, and availability are generally considered security requirements [13]. User/patient privacy, customer behavior, message authentication, and control messages are the most important security requirements before sensors are deployed [14]. The security of cryptography keys is a crucial factor in ensuring the confidentiality and integrity of data. To maintain the security of cryptographic keys, health systems must handle secure key management for many devices. A number of studies have been conducted regarding key management systems in recent years. The topics discussed in existing studies on WSNs include architectures, applications, communication, and cyber security. Some studies, on the other hand, dealt with key management systems for WSNs, a very critical area of research that has received surprisingly little attention. In these studies, lightweight encryption was highlighted in addition to a key management scheme that achieves a defensive mission in resistance to WSNs threats [15]. Therefore, an efficient energy-aware secure key management method is significant. The key distribution scheme in a WSN has to satisfy some objectives, such as a low memory requirement, low overhead for computation and communication, and high connectivity and robustness. Managing key information and data delivery in the network is another issue that needs to be addressed. Keys of the same length as the message can be consumed by the encryption process in private key management systems. The key is consequently a limited resource in a WSN. A network with several communication activities will be inefficient and unstable if there is no key management [16]. It should use methods from traditional network research that handle issues such as these to tackle this challenge and improve network activities management.

With the growth of the global population, as remote health monitoring becomes more popular, the demand will increase extremely in the coming years. A major goal of remote health monitoring is to transfer patient information to clinical physicians across the globe [17]. The importance of securing patients' clinical information in this scenario grows, so that unauthorized individuals cannot alter or read it. In terms of encryption, Rivest Cipher (RC5) is a simple and secure cipher. For limited resources environments, such as WSNs, it is considered a suitable block cipher because of its simplicity, fast encryption, low power consumption, easy adaptability and low memory requirements. Key calculation with RC5 is susceptible to attack because of its weak diffusion state. By using RC5 in combination with key management and randomness generators, this can be overcome so that medical data can be ciphered and protected [18] while preventing classifications of node capture attacks.

1.1. Major Contributions

To address all previous issues, we propose a reliable method based on energy-saving routing, lightweight encryption and randomization techniques to achieve efficient key management for WSN. First, the HEED protocol is adopted to support efficient routing and sensor energy conservation, thus supporting energy saving as much as possible to extend the lifetime of the WSN. Second, we use a lightweight RC5 encryption algorithm to maintain medical environment data. Third, we generate unique randomness using PRNG to support RC5 and prevent the attacks from breaching the encryption. Finally, we utilize the Bloom scheme to manage the keys in a secure manner that protects medical environment data. All these techniques are integrated into a single security method to protect patient/provider data and information.

1.2. Research Organization

A description of the research roadmap can be found here: A comprehensive introduction is provided in Section 1. We critique related key management security works in Section 2. The requisite preliminaries are introduced in Section 3. Our proposed method is described in Section 4. Section 5 investigates the proposed method results. Section 6 presents the study's conclusions and future trends.

2. Schemes Related to the Security of WSN Key Management

In this section, we will investigate key management methods and extract their problems and drawbacks.

For a heterogeneous WSN, Li and Wang [19] proposed an effective and hybrid key management strategy. While symmetric methods were used between the cluster's sensors, elliptic curve cryptography was used to generate the key between the cluster heads and the sink. A low-cost, high-level security authentication and key management scheme (AKMS) was intended to be provided as protection from hostile sensors that can appear during networking. Even if the AKMS keys are compromised, attackers cannot utilize the prior keys or the authenticated sensors to cheat. In particular for heterogeneous networks, simulation findings demonstrate that their approach offers effective security with decreased energy usage. However, their scheme is not very safe against cluster head capture attacks. The network model is hierarchical, according to Iwendi et al. [20]. Both the pairwise key between the cluster heads and the base station and the key between sensors and their particular cluster heads have been generated in a symmetric manner employing OR and XOR operations. The approach provides security and makes inefficient use of limited resources, but it also lacks scalability. Zhang and Pengfei [21] purposed approach to secure hierarchical network structures. This method made use of three different sorts of keys. In the first stage, a disposable paired key was formed using the specified function for use in encrypting data exchanged between nodes, and the primary keys were generated using Diffie-Hellman and the specific function. However, the authors do not address issues of secure key storage. Furthermore, their scheme is not suitable for medical environments that require reliable key management and lightweight data encryption. Zhang and Wang [22] suggested a key management method in hierarchical WSNs based on a Bloom scheme with sophisticated advanced encryption standard (AES) and a mesh module for multi-hop packet routing, with high security and scalability. However, their scheme did not provide a mechanism to support random health data encryption. Qin et al. [23] have developed a hybrid key management system (KMS) for multihop WSNs that makes use of secret key-based communication and asymmetric cryptographic approaches to minimize the computational burden on member nodes. KSM's security analysis demonstrates its ability to resist node capture attacks and support node revocation. Data freshness, the number of generated typical keys, throughput, and cost of computation were used to assess KSM's effectiveness. However, their scheme did not provide updating keys for the hierarchical medical WSN.

On off-the-shelf static WSNs, Moara-Nkwe et al. [24] discussed challenges and difficulties experienced during the establishment and application of physical layer secure key generation (PL-SKG) methods. It then suggested a method for generating keys using elliptic curve cryptography (ECC) based on signals from 802.15.4 compliant sensors that could take advantage of the power, simplicity, and diversity of frequency channels available. However, generating keys using asymmetric encryption algorithms will add computation and communication costs to the WSN environment. A key management protocol presented by Chanda et al. [25] is claimed to guarantee the confidentiality, integrity, authenticity, and integrity of wireless sensor networks by handling key generation, distribution, and maintenance. Their proposed method encrypts network information in three levels using three auxiliary keys in addition to the main key. Unfortunately, their protocol fails to provide unique and sufficient keys to protect WSN data and their method is very complicated and resource-consuming for WSN. A network model for the intelligent building energy management system (IBEMS) was developed based on the framework of the WSN [26]. Then, the IBEMS presented a blockchain-based dynamic key approach as well as key management, examining the security of blockchain technology with the Shamir scheme. Experiments were conducted to verify their plan's feasibility. However, the authors did not provide a clear key management technique, they relied on Shamir's secret sharing for key exchange but did not specify the threshold in their method.

Ahlawat and Dave [27] proposed a secure hybrid key pre-distribution scheme (HKP-HD) for WSNs in order to prevent node capture attacks. By combining q-composite and threshold-resistant polynomial schemes, they claimed robustness. Their scheme investigated to make the WSN more solid against the sensor capture threats. There is a presumption that hacker is intelligent and that they frequently develop a matrix of attacks against the network by taking advantage of various weaknesses. It attempts to destroy the whole network with the fewest possible sensors, based on the attack matrix. In order to counteract such vulnerabilities, a comparable threat array was created by the network engineer by investigating sinks as major influencing factors. However, their method was only seeking to decrease the risk of keys being compromised and not to end the problem completely, and this in itself is a security breach. Kumar and Malik [28] examined the keys required to develop resilient and connected WSNs that have a large number of sensors. An improved random key distribution method based on random deployment was presented to increase connectivity and resilience. For the large, medium, and small-scale networks, they investigated the number of keys that are sufficient. However, they did not use a routing protocol to reduce power consumption in WSNs. Recently, Tyagi et al. [29] discovered several security pitfalls in previous methods, such as a man-in-the-middle, an off-line password guessing, and session key attacks. An Internet of Things (IoT) authentication method was created to overcome the pitfalls identified in previous methods. Furthermore, a real-or-random (RoR) model was used to confirm the reliability of their method. Based on computation and communication costs as well as security properties, they evaluated their proposed method against the associated schemes. However, although the authors claimed that their method provides key protection, their method did not provide key security management. Furthermore, although the [30,31] tested their proposed methods against node capture attacks, their approaches are complex and inflexible in handling sensor-transmitted parameters in medical environments.

3. Introductory Details of the Proposed Techniques for the Security of WSN Key Management

In this section, we will outline the fundamental concepts behind the techniques employed in the proposed method.

3.1. Hybrid Energy-Efficient Distributed Clustering Protocol

A big crowd of WSN routing approaches addressed the energy conservation issue. Hybrid energy-efficient distributed clustering (HEED) [32,33] and low-energy adaptive clustering hierarchy (LEACH) [34,35] are the most distinguished hierarchical routing-based WSN protocols. However, there is a negative impact on the network's cluster heads' (CHs) in LEACH distribution when carrying out rounds [33]. In addition, the comparison in Table 1 demonstrates that the HEED protocol outperforms the LEACH protocol [36].

Table 1. Comparison of performance properties between HEED and LEACH protocols.

Properties	HEED	LEACH
Balanced clustering	Good	Moderate
Balanced loading	High	Moderate
CH capability	Data aggregation, homogeneous	Data aggregation, homogeneous
Clustering process execution	Iterative	Probabilistic
Cluster overlapping	No	No
Cluster stability	High	Moderate
Delay	Moderate	Very small
Energy efficiency	Moderate	Low
Mobility	Stationary	Stationary
Routing between clusters	Single hop and Multi hop	Single hop
Routing within a cluster	Single hop	Single hop
Scalability	Moderate	Low

HEED protocol maximizes network lifetime by reducing communication costs and utilizing residual energy in sensors. In a set number of iterations, HEED completes the clustering phase, creates well-distributed CHs, reduces control overhead, and optimizes network lifetimes. Sensor distributions or sensor density in a network do not affect HEED [32]. Due to the fact that new CHs are always chosen and clustering starts after each interval of the clustering process time (T_{CP}) + operation time (T_O). Receiving and transmitting messages from neighboring sensors within a defined range is a time-consuming process. HEED defines a fixed percentage of CHs in order to begin clustering. Initial CHs probabilities are set by sensors according to the formula:

$$CH_p = C_p \cdot (E_r / E_m) \tag{1}$$

An initial probability, residual energy, and maximum energy of the sensors are represented by C_p, E_r and E_m, respectively. In order to meet minimum probability (Pmin) = 0.0001, CH_p must not fall below Pmin. Figure 1 shows the clustering approach in HEED protocol.

Sensors periodically communicate with their neighbors about their current status during each round. When sensors identify themselves as CHs or receive an invitation to join from another CH, they are regarded as covered. If a node is running HEED but is still visible, it should declare itself a CH or join the neighboring cluster. As part of the HEED protocol, wireless sensor networks are organized into clusters. An elected sensor from each cluster gathers raw data from its associated sensors and transmits it to the sink. As soon as the sensors for level-1 have been chosen and the HEED protocol has been employed to cluster the network [37]. By reusing the HEED approach, super-elected sensors (level-2) are in this situation elected using a larger cluster radius. A second run of the HEED approach will result in the network being divided into two categories of clusters. An elected sensor of level-1 is part of a cluster comprised of regular sensors within a radius transmission range (Tr1). Super-elected nodes receive data collected from regular nodes within the cluster. After the second HEED protocol execution on the level-1 chosen sensors, the second category of clusters is established. It is made up of a cluster of elected level-1 sensors that are placed close to the Tr2 cluster and an elected level-2 sensor that is in charge of receiving information from the cluster's various members (elected level-1 sensors) and sending it to the sink [37].

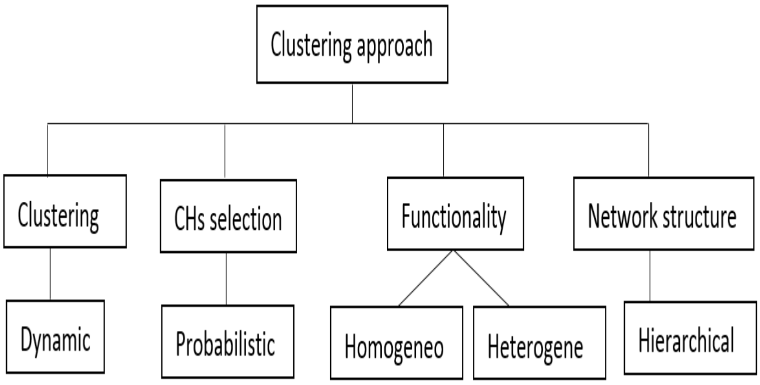

Figure 1. HEED approach of clustering.

3.2. Rivest Cipher 5

Ronald Rivest proposed Ron's code or Rivest cipher (RC5) in 1994. This cipher uses block ciphers of symmetric type and is fast. This cipher can be implemented in both hardware and software. Rotation based on data is extensively used in RC5. Both linear and differential cryptanalysis can be prevented by this feature. In this algorithm, the block size and round numbers are parameterized, as well as the key length. As a result, both performance and security are greatly enhanced. A specific RC5 algorithm is the word/round/byte (w/r/b) algorithm. The w bit size is 16, 32 (standard value) and 64. Because RC5 encodes two-word blocks, both plaintext and ciphertext are two words long. Moreover, r values are (0–255), and table (t) = 2 words are included in the expanded keys table. In addition, the number of bytes (b) with values ranging from 0 to 255 specifies the security key. Encryption, decryption and key generation are the three elements of RC5 [38].

A comparison of RC5 with Rivest-Shamir-Adleman (RSA) and Blowfish shows that it is more secure and faster. Sharing secret keys securely remains a challenge with RC5 since it is a symmetric key cryptosystem. This limitation was overcome by combining RC5 encryption with Honey encryption, which had a bigger buffer size and maintained RC5's strengths at the same time [39]. There are three block sizes for encryption: 32 bits, 64 bits, and 128 bits. The best block size is 64-bit. RC5 keys range from 0 to 2040-bit, but 128-bit is most commonly endorsed. Plaintext and ciphertext are stored in two 32-bit registers (A and B). Normally, encryption takes 12 rounds (but it can take as many as 255) [40]. Figure 2 shows RC5 process. In RC5, key operations involves XORing bits, adding words modulo 2w, and shifting left (<<) and right (>>). Due to its flexibility in terms of key size, block size, and rounds, RC5 offers high levels of security and performance.

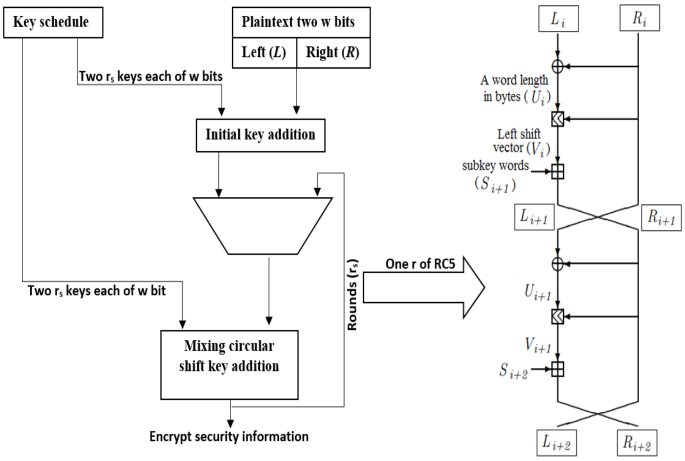

Figure 2. RC5 process.

3.3. Bloom Scheme

To determine whether an element is part of a set, a space-effective probability data structure called a Bloom filter is utilized. Essentially, the matrix starts out with all bits set to 0, it is a bit matrix of length n. A Bloom filter employs k distinct hashes $\{h_1, \dots h_k\}$ with a range of [0, n 1] to exemplify a set $S = \{x_1, \dots x_m\}$. The Bloom filter's bits $h_i(x)$ are set to 1 for each element $x \in S$. Multiple instances of setting an index to 1 have no impact; only the initial change does. It is necessary to determine whether all positions of $h_i(x)$ are set to 1 in order to determine whether an element y is in S. Despite the fact that this technique is quick and effective, it is possible to obtain false positives if the bits were accidentally changed from 0 to 1 during the intercalation of another element y where $y \in S$ and y *neq* x. Figure 3 [41] provides a diagrammatic representation of the general structure of a Bloom filter.

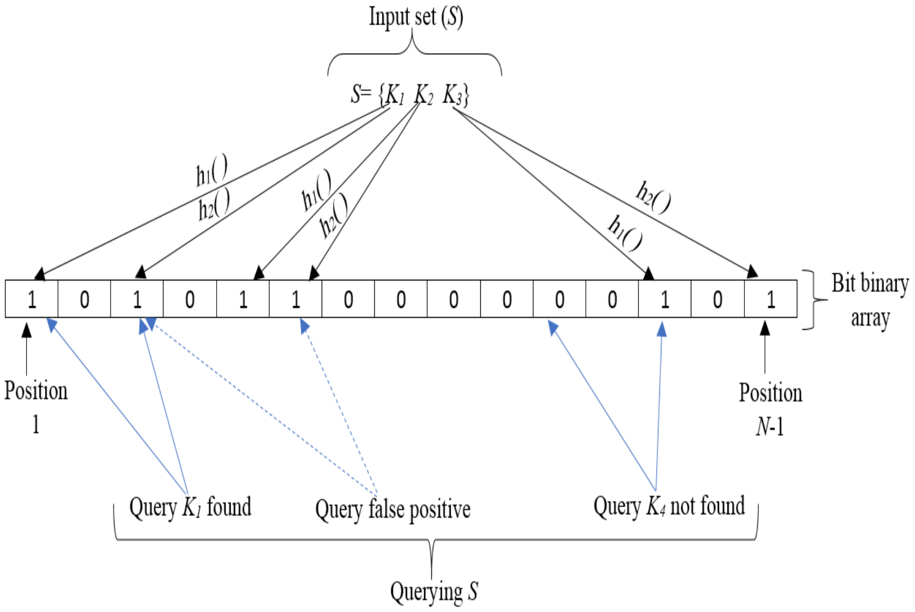

Figure 3. Bloom process.

Bloom filter-based schemes can be made more secure by using a randomly generated key instead of the same key for every filter. Additionally, it is recommended that the key length be at least as long as the Bloom filter, which is the second and most crucial condition. Since absolute secrecy can only be used in specific circumstances, it is crucial to include these qualities. A health biometric protection system based on the Bloom filter that concurrently meets important security needs such as irreversibility and unlinkability with other desired qualities such as recognition effectiveness and data compression. Due to its simplicity and lack of a necessity for pre-aligning the biometric templates, the Bloom scheme quickly gained popularity. Some threats relied mostly on the fact that the encoded templates were somehow tied to the original health biometric information. In order to develop some attacks, it was observed that the original biometric template and the encoded template had the same hamming distance. To reduce extra inputs and outputs induced by checking multiple tables, modern designs use Bloom filters in repositories stores to quickly check the existence of a key pair in an individual table.

The IDs of revoked certificates can be fed into a Bloom filter to condense the revocation list. In probability theory, Bloom determines whether a given element belongs to a set. However, the member's query either returns "possibly in the set" or "definitely not in set", demonstrating the possibility of the Bloom filter finding false positive results. The bit vector for the Bloom filter has a length of m bits and is initially commenced to zero. The certificate serial number $Sensor_i$ is kept in the bit vector for certificate revocation list (CRL) compression after being hashed using the k-hash algorithms. In order to save the element $Sensor_i$, all addresses in the m bit vector that are pointed by the K hashes of the certificate are set to one. The position of the prepared Bloom is compared with the hashed location of the given $Sensor_i$ of the certificate in order to validate it in the vector of the Bloom filter. It is possible that the given $Sensor_i$ of the certificate is on the list if all bit locations are set (matched), otherwise, it is not. In spite of this, it is possible to set the bit to one multiple times since different hashes may point to the same place. The bit vector is made up of the hashes of various $Sensor_i$ for the certificates. As a result, a false match occurs, and the false positive rate is computed as follows. The chance that the location Bi is set to one is given by $(1 - (1 - 1/m)^{KN})^K$, where

$$P(FalsepositiveRate) = (1 - (1 - 1/m)^{KN})^K \tag{2}$$

Consequently, a non-revoked certificate could be interpreted as revoked, which could cause the search to return that is inappropriate for the filter [42].

3.4. Pseudorandom Number Generator

In wireless networks with limited resources, such as WSN, pseudorandom number generators (PRNGs) are a well-liked option for cryptographic methods for key generation. This is partly because of their capacity to produce distinctive sequences from various seeds. Furthermore, these generators can produce long-period sequences devoid of repetitions. For generating key sequences in radio-frequency identification (RFID)/WSN applications, such algorithms have also been considered. In order to assure bit dispersion in the pseudorandom sequence, these PRNGs might include nonlinear filter functions or use different feedback polynomials. However, it should be emphasized that PRNG-based techniques only aid in key generation and management; for authentication, additional methods, such as hash-based or trusted third-party-based procedures, should be used in conjunction with them [43].

Strong foundations are necessary for the existing key management strategies. This might be carried out by improving the basic random number generation procedure that the BS uses to generate initial random numbers. In addition, key randomness techniques based on PRNGs have shown strong initial energy-efficient performance for IoT nodes [44], particularly in health systems. There are several forms of PRNGs that can be applied to clients' health applications. Linear feedback shift registers (LFSRs) are widely utilized as cryptographic primitives, stream ciphers, PRNGs . . . etc. because they are highly simple, effective, and reasonably quick circuits. However, because LFSRs are linear, predictable, and dependent on strong seeding, they introduce flaws that have been exploited in previous systems. The Mersenne Twister is another well-liked PRNG since many programs packages use it as a standard PRNG. It has a very long time before repeating since Mersenne Twister relies on the Mersenne prime ($2^{19937} - 1$). It is a highly quick and effective PRNG, which leads to its acceptance by many software platforms, including MATLAB, Java, and Python.

With the use of the National Institute of Standards and Technology (NIST) test suite, the unpredictability of these various PRNG stream outputs was evaluated [45]. To show that shorter LFSRs often do not produce better randomness than longer ones, two different-length LFSRs were investigated in the previous method. The 15-tap LFSR failed a number of the tests as was to be expected, demonstrating its lack of security as a PRNG. The Mersenne Twister came extremely near to passing one of the tests, but it did not achieve the minimum acceptable pass rate (98.65% vs. 98.7%). The performance of the PRNG scheme was then assessed using session key streams produced using the same PRNGs. A stream of 80,000 128-bit was created with each PRNG using a random 1024-bit (providing over 10 million bits for statistical testing per PRNG). The NIST test suite was then used to verify the randomness of the produced bits. The weaker LFSR's subpar results were effectively concealed by the PRNG scheme, and all PRNG tests passed. Following the execution of these two test cases, more thorough NIST statistical tests were conducted to look for further trends and defects in the output. All PRNGs worked Absolutely fine, however, the 15-tap LFSR failed. Therefore, using the security scheme, any PRNGs other than the 15-tap LFSR are considered to be adequately safe. An additional layer of security is provided on top of the PRNG depending on the size of the window utilized and the initial starting index (both internal states of the security scheme), which should be preserved safely in the case of a PRNG compromise. A robust PRNG, such as a cryptographically secure PRNG (CSPRNG), however, would still prevent an attacker from generating a correct derived key even if the internal state of the security scheme were exposed [45].

4. The Proposed Method

This paper investigates protecting a heterogeneous WSN in which the sensors have limited capacities and are clustered in diverse ways based on the HEED protocol. Each cluster has a CH who is in charge of the member nodes' communication and collecting the information. The member node performs one task and transmits the information from the surrounding area to the group's leader. Each connected party in the network has to have a secure link in order to protect the transferred information. It should share the secure key to perform cryptographic activities and meet security requirements. Therefore, the proposed method includes key generation, encryption and decryption procedures, key updating and sensor add/delete.

4.1. Key Generation

Prior to placement in the target health area, each sensor is pre-loaded with a key, PRNG, and unique identifier. After the deployment phase, clustering is carried out utilizing the HEED protocol, as previously indicated. The key between each connected node should now be established. The key between neighboring cluster heads and the cluster head and the BS is managed by applying the Bloom scheme, which is utilized in our proposed method in an efficient manner that maintains WSN resources, PRNG is used to generate keys between cluster heads and member nodes.

4.1.1. Key Generation between CH-CH and CH-BS

The Bloom scheme provides high security, and a low amount of overhead achieves a balance in the use of each node's resource and provides scalability. As the standard matrix is a square matrix with zeros and ones to simplify the components similar to the columns, we employed an adjacency matrix to decrease processing and storage. All sensors that are a specific sensor's neighbors are filled with ones in this adjacency matrix, while the other elements are provided with q-1 so that they cannot contain zeros. The adjacency matrix lowers storing the columns in the node's memory. Subsequently, any node can build an adjacency matrix. As Bloom's approach, the prime number is important to produce the keys, which number depends on the required key length. The following array shows the adjacency matrix in its original binary form.

$$\begin{vmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{vmatrix}$$

The steps for computing the key are shown below.

- **First step:** We select a prime element from the field GF(q), where q is bigger than the key length and $q > N$. Next, our method constructs a public matrix G based on the sensor neighbors that is $N \times N$ in size relying on the λ, value a number of rows with N columns.
- **Second step:** The BS produces a D symmetric matrix of size (λ + 1) × (λ + 1). Then, it computes matrix A by $A = (D \cdot G)^T$.
- **Third step:** All rows of matrix A stored in a memory of the sensors. When sensor i wishes to connect with sensor j, sensor i multiplies the row A_i with the column G_j. Then the result is a secret key. To demonstrate the operation of the modified Bloom's method using an adjacency matrix. For example, the network has 6 sensors,

for instance, N = 6, $lambda$ = 3 (secure parameter), and q = 29 (prime numbers). Modified adjacency matrix:

$$\begin{vmatrix} 28 & 1 & 1 & 28 & 28 & 28 \\ 1 & 28 & 28 & 1 & 28 & 28 \\ 1 & 28 & 28 & 1 & 1 & 28 \\ 28 & 1 & 28 & 1 & 28 & 28 \\ 28 & 28 & 1 & 28 & 28 & 28 \end{vmatrix}$$

Public matrix (G):

$$\begin{vmatrix} 28 & 1 & 1 & 28 & 28 & 28 \\ 1 & 28 & 28 & 1 & 28 & 28 \\ 1 & 28 & 28 & 1 & 1 & 28 \\ 28 & 1 & 28 & 1 & 28 & 28 \end{vmatrix}$$

Secret semantic matrix (D):

$$\begin{vmatrix} 3 & 5 & 2 & 7 \\ 5 & 6 & 9 & 1 \\ 2 & 9 & 3 & 5 \\ 7 & 1 & 5 & 4 \end{vmatrix}$$

A= $(D.G)^T mod$ 29:

$$\begin{vmatrix} 26 & 9 & 5 & 24 \\ 3 & 20 & 24 & 5 \\ 18 & 18 & 14 & 26 \\ 22 & 20 & 28 & 14 \\ 16 & 26 & 16 & 22 \\ 12 & 8 & 10 & 12 \end{vmatrix}$$

To suppose two nodes such as sensor 2 and sensor 5, who wish to communicate with one another, we shall multiply sensor 2's private row from matrix A, which is A (2) in sensor 5's public column, by G. (5). In a similar manner, sensor 5 multiplies its private row A (5) in node 2's G public column (2). The previous operation will generate the shared secret key for sensors 2 and 5.

$$K_{5,2} = A_5 \cdot G_2 = |16 \;\; 26 \;\; 16 \;\; 22| = \begin{vmatrix} 1 \\ 28 \\ 28 \\ 1 \end{vmatrix} = 1214 \; mod \; 29 = 25$$

$$K_{2,3} = A_2 \cdot G_3 = |3 \;\; 20 \;\; 24 \;\; 5| = \begin{vmatrix} 28 \\ 28 \\ 1 \\ 28 \end{vmatrix} = 808 \; mod \; 29 = 25$$

It used the initial key to cluster-head for encrypting any row of the A matrix. The row for a certain cluster head and key ID should transmit. CBC-RC5 is the encryption method utilized. CH will receive a message with its row and key ID and will work to decrypt it before storing it in the sensor's memory. Now each cluster head has its own unique row and key ID. It should be formed as the shared key between CH-CH and BS-CH. Several of CHs directly communicate to the sink over a single hop and others are not directly connected to the sink but through neighboring cluster-head, allowing them to broadcast data across them until they reach the base station. The shared key is calculated as follows:

Shared key N_i = Row of Node i * Public column of Node j;
Shared key N_j = Row of Node j * Public column of Node i.

4.1.2. Key Generation between CH-Sensors

Before deployment, each sensor node was pre-loaded with an initial-key that was utilized to form a key between the CH and the member sensors. The CH and member sensor both through the suggested PRNG generate a shared key, and the authentication key is derived from the shared key using the PRNG. Figure 4 shows the PRNG process to generate the shared key. First, we divide the input initial key value into four parts (K_1, K_2, K_3 and K_4). Second, we use a set of variables X, Y, Z, Q, T, V, F and U and a set of operations such as XOR, not, addition and left shift (<<) to obtain high randomness and then store the random result in four registers A, B, C and D to obtain the shared key of 64 bits.

- Step1: The initial key has been split into four parts K_1, K_2, K_3, and K_4

```
1.  For $j$ from 1 to 32
    –  $Z$ [Bit XOR($K_1, K_3$)]
    –  $Y$ [Bit XOR($K_2, K_4$)]
    –  For $u$ from 1 to 16
    –  $V$ [swapping($Y$) $<< 5$]
    –  End
    –  $X$ addition($Z$,$V$) module $2^{16}$
    –  $T$ [Bit XOR($Z$,$Y$)]
    –  $Q$ [Bit XOR($V$,$X$)]
    –  For $i$ from 1 to 16
    –  $U$ [swapping($Q$) $<< 9$]
    –  end
    –  $F$ addition($T$, $U$) module $2^{16}$
    –  $a_1$ [Bit not($X$)]
    –  end
2.  For $j$ from 1 to 16
    –  For $s$ from 1 to 16
    –  $b_1$ [Bit not($V$)]
    –  end
    –  $c_1$ [Bit XOR($V$, $F$)]
    –  For $n$ from 1 to 16
    –  $d_1$ [Bit not($F$)]
    –  end
    end
    $A$ - [binary Vector to Hex($a_1$)]
    $B$ - [binary Vector to Hex($b_1$)]
    $C$ - [binary Vector to Hex($c_1$)]
    $D$ - [binary Vector to Hex($d_1$)]
```

- Step2: The four registers [A,B,C,D] combine to form the final key (shared key-64 bit).

4.2. Encryption and Decryption Procedures

For the sake of providing high security, the security requirements (confidentiality, integrity, and authentication) should be met. A combination of CBC-RC5 is employed in this work to do this task at once as shown in Figure 5. In order to protect the shared key (SH-K), our proposed method inserts this key (generated from the previous random PRNG process) into the CBC-RC5 algorithm for encryption, as this algorithm is known for its ability to block analysis, differential and node capture classifications attacks in addition to its speed in encryption. Hence, the output of this algorithm is the Auth-K key which is used to securely protect the shared key of the health sensors (CH-Sensors).

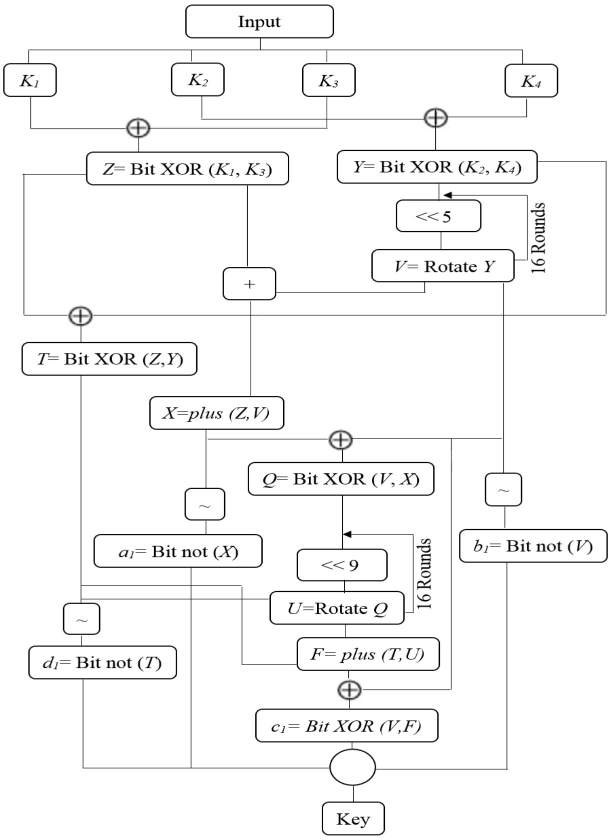

Figure 4. PRNG process with a shared key.

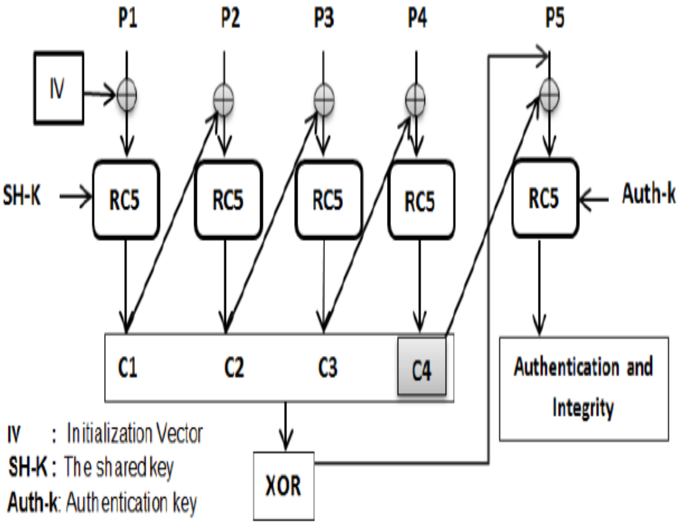
P1
P2
P3
P4
P5
IV
SH-K
RC5
RC5
RC5
RC5
RC5
Auth-k
C1
C2
C3
C4
Authentication and Integrity
XOR
IV : Initialization Vector
SH-K : The shared key
Auth-k: Authentication key

Figure 5. Diagram of CBC-RC5.

4.3. Key-Updating

The key must be updated on a regular basis after time has passed to prevent the hacker from having access to existing key data. The sink transfers a new row and *K*-id encrypted by initial-key to the CH. In this case, our proposed method changes the shared key between CH-CH and CH-BS regularly (see Section 4.1.1). The PRNG is used in conjunction with the previous Auth-*K* to update the Auth-*K*. Additionally, the shared key between CH and sensors needs to be updated. The existing shared key between CH-sensors is used in the PRNG (see Section 4.1.2) to create two new keys: a new SH-*K* and a new Auth-*K* between CH-Sensors.

Sensor Add and Delete

The new sensor can be declared as a cluster head or linked as a member node to another cluster head. If the new sensor becomes a CH, it transfers a need data to the BS, which the sink responds to with a row and key-ID for the new sensor. Then, as mentioned in the key establishment phase, it will generate a shared key. If the additional sensor becomes a member sensor of one CH. To authenticate the new sensor and obtain the new member's initial-key the CH transfers data to the sink, after which the shared key is generated as previously mentioned. If any sensors fail or become compromised, the sink sends out messages to all sensors in the network, instructing them to eliminate the node's ID from the nearby table.

5. Results

This section will explain the performance and security results of our proposed method. Our proposed method focuses on the use of heterogeneous WSN in medical environments with static locations for sensors because firstly these environments are important for people's lives, secondly, the use of this proposal may not be suitable for other environments such as military, natural phenomena such as earthquakes . . . etc. which depend on random distribution of sensors, thirdly, the use of a static distribution of sensors in medical environments makes it easier for us to evaluate performance accurately and without fluctuations.

5.1. Performance Results

The proposed method depends on the distribution of 100 nodes in an area of 100 m × 100 m where the position of the BS is (50.50). The nodes consist of two types: 80 nodes have low resources (0.5 joules of energy, 25 bands, low compute capacity), and 20 nodes have higher resources (2 joules of energy, 40 bands, low compute capacitance). Figure 6 depicts WSN in our proposed method. The distribution of sensors in our method depends on the static distribution because it is applied in a healthy environment. Furthermore, MATLAB 2020b was used to perform the simulation under Windows7 64 bits operating system with CPU i5-2540M @ 2.60 GHz and RAM 4.0 GB. Examining the performance of our method is very important but it is very difficult to find similar methods to our method under the same conditions and parameters. Therefore, we tried to find the closest existing method and compare it with our method to prove its superiority and acceptability. Our proposed method is compared with the two existing methods: A highly dynamic secret key management (DSKM) [21] method and a smart security implementation (SSI) [22] method for WSN nodes under the same circumstance. Figure 7 displays the clustering stage utilizing the HEED approach. This figure shows the correlation of the sensors to the closest CH based on the HEED method.

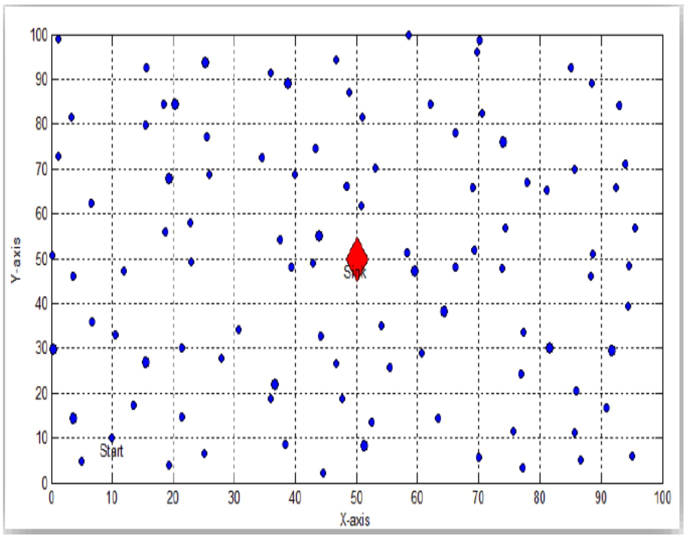

Figure 6. Random-distribution of the sensors.

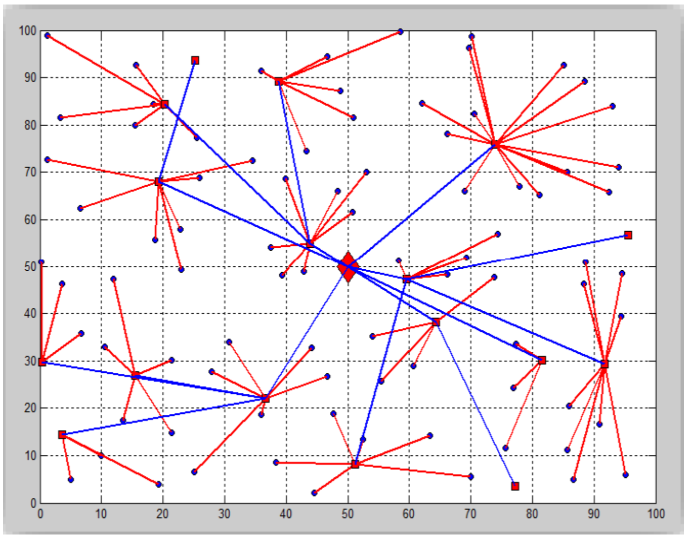

Figure 7. The clustering stage.

Figure 8 displays the energy expendable for generation shared keys comparison with the DSKM and SSI methods. The result is that our method expends less energy than the others. Where we note that our method performs significantly better than SSI in energy conservation and slightly better than DSKM, this means that the sensors in our method will collect health data for a longer period.

Figure 9 displays the size of memory used in cluster heads. In terms of memory expenditure in CHs, we notice in this figure that SSI is less memory expenditure compared with our method and DSKM but DSKM suffers from the different and unstable fluctuation of memory expenditure, generally, our method is relatively stable in memory usage compared with SSI and DSKM.

Figure 10 displays the size of memory used in the member sensor. In terms of memory expenditure in sensors, our method has a size of memory used less than precedent methods, which require twice the amount of memory. Figure 10 shows that SSI and DSKM are very memory-consuming compared to our method. Where we notice that the sensors in SSI are very memory-consuming compared to DSKM and our method. However, this figure shows that the health sensors in our method do not require large memory expenditures because we use lightweight techniques to generate shared keys and authentication keys. Figure 11 displays the processing time for generation shared keys. In comparison to the two existing methods, our proposed method to generate keys is lightweight. The use of HEED, RC5, Bloom and PRNG achieves fast and lightweight operations, which makes the processing time of our method very fast compared to SSI and DSKM. As we notice from Figure 11 that DSKM requires a very large processing time compared to SSI and our method. Finally, we note that our method is superior to SSI and DSKM in terms of energy consumption, memory expenditure in CHs, memory expenditure in sensors and processing time. However, there are some limitations to the proposed method. First, if the sensors are randomly distributed (in environments other than healthy ones), the results may differ. In a healthy environment, we can put the sensors in static locations, which provides the ability to control the stability of the results, but if they are used in a different environment, for example, the military, which requires random distribution, which may lead to different results. Data size and key sizes also can affect the results (it is left for future work). Duplicate data/information or decryption without detection of the encryption breach could consume WSN resources which are not addressed in this proposal.

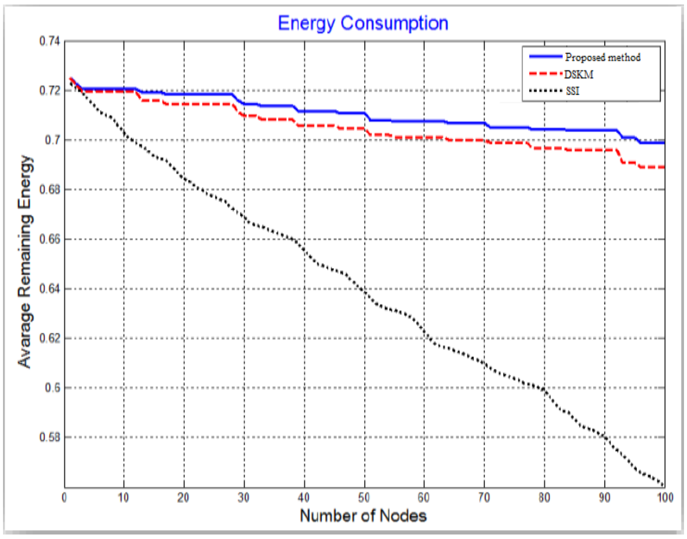

Figure 8. Energy consumption test.

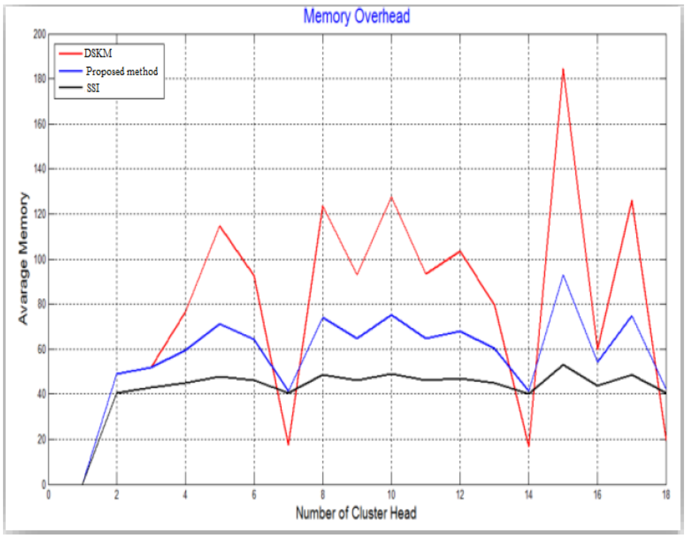

Figure 9. Size of memory used in cluster-heads.

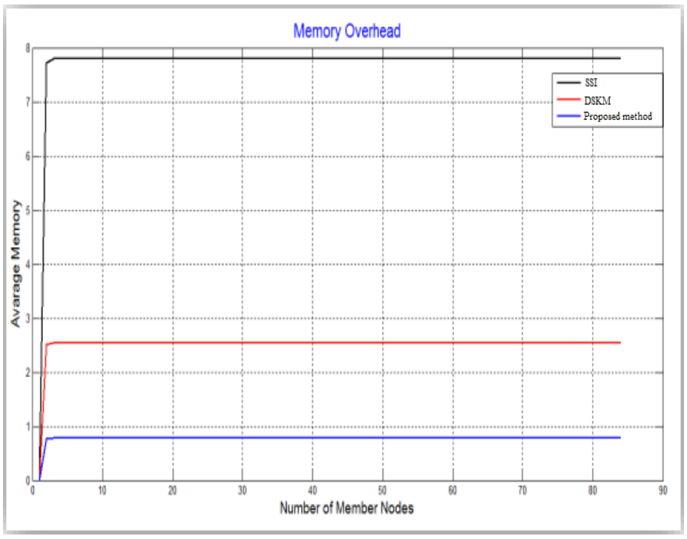

Figure 10. Size of memory used in the member sensors.

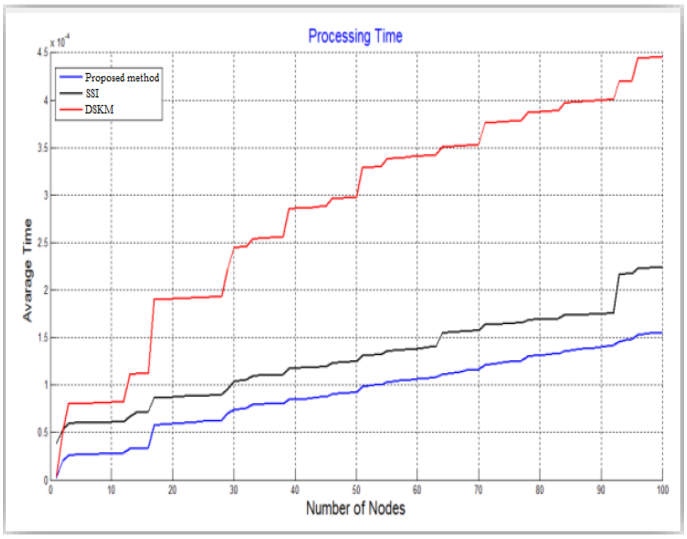

Figure 11. Processing time for generating shared keys.

Theoretically, a comparison of some recent references with our proposed method shows the superiority of our proposal in terms of performance. Refs. [24,25] used asymmetric cryptographic algorithms such as ECC for the key generation which will add significant costs to WSN resources while our method is based on PRNG which is a lightweight key generation method. In addition, in [26,27], the authors did not specify a suitable method for routing information, parameters, and sharing keys within the sensor network while our method relies on HEED which provides a suitable routing for sensor energy conservation. Moreover, the ref. [28] depends on randomly distributing the sensors, this leads to different distances between the sensors which will lead to a significant increase in the sensor communication expenses, this problem is avoided in our proposal due to the static distribution of the sensors in the medical environment. Finally, the ref. [29] relies on hash-intensive operations as well as encryption operations in key management which will negatively affect the computational costs of the sensors while our method relies on Bloom-lightweight key management.

5.2. Security Results

The attacker attempts to use node capture attacks to compromise information and WSN parameters. Five classifications of capture attacks are possible: sensor node, sensor CH, BS, and more than one sensor node or CH. These attacks try to penetrate shared keys, authentication keys and availability. The flaws that were exploited by capture attacks include problems with dictionary and forward secrecy, improper parameter distribution, irrational design purpose, ineffective verification, and unsecured parameter communication.

- Sensor node capture attack: When the attacker compromises a sensor and obtains some previous parameters such as SK-K, they try to use that key in future sessions of the WSN. In our proposed method, all sensors use new SH-K for each session. Therefore, when a hacker performs a sensor node capture attack on our WSN it will not affect the confidential information of other sensors.
- Sensor CH capture attack: When the hacker succeeds in executing this attack on CH. It tries to use the previous Auth-K to make all its sensors trust it and send all data and information to that hacker in that session. In our proposed method, the sensors within the cluster do not handle the old Auth-K. Thus, this attack cannot compromise WSN information by relying on a single CH, namely, our proposed method resists the CH capture attack.
- BS capture attack: We assume that BS is safe against capture attacks. However, assuming that the hacker was able to penetrate the BS either remotely or by stealing the BS device. The hacker will not benefit from the previous information of sensors or CHs because all security parameters (such as SH-K and Auth-K) in our proposed

method are generated instantly/unique by PRNG and Bloom and are hidden by RC5. However, the hacker may find some data collected by sensors. First, we assume that the data is transferred periodically to a central server so that even if the hacker tampered with this data, the original copy will be safe. Second, our research focuses on security key management and not the data collected. Therefore, our method is able to block BS capture attacks.

- A capture attack of more than one sensor node: If the hacker was able to compromise two or three sensors. Then he tried to analyze the obtained security parameters (such as Auth-*K*s) for these sensors. The hacker cannot use these current parameters to hack network information in the current session. Because our method uses the RC5 algorithm, which has the advantage of preventing analysis and differential risks. Therefore, our method prevents this attack from extracting security parameters from Auth-*K*s.
- A capture attack of more than one CH: When a hacker can compromise two CHs or three CHs. It tries to use the security parameters available from the compromised CHs. However, our proposed method uses a Bloom filter between BS and CHs to manage and verify the exchanged keys. The hacker cannot use the old parameters to communicate with the BS because these parameters will be rejected by the BS. Therefore, our proposed method is able to prevent this attack.

Table 2 shows the comparison of security features between the proposed method and the security key management methods in WSN. Where Sym is symmetric encryption and Asym is asymmetric encryption. The [23,27,30] methods are not discussed for classifications of node capture attacks. This indicates that their methods can be an easy target for various classes of node capture attacks. While our method and ref. [31] investigated different classes of these attacks. However, ref. [31] did not discuss compromising multiple sensors and CHs, nor did they specify countermeasures. Moreover, our method provides high randomness (by using PRNG) to the shared keys which is superior to existing methods that use low or medium randomness. The high randomness gives Auth-*K*s and SH-*K*s keys resistance to analysis and deferential threats. Finally, our method uses a flexible manner such as the Bloom scheme to manage security keys where Bloom is not used in the existing methods.

Table 2. Comparison of key management methods with our proposed method.

Security Feature	Qin et al. [23]	Ahlawat and Dave [27]	Liu et al. [30]	Wang et al. [31]	Proposed Method
Anti node capture attacks	One	One	One	Many	Many
Encryption type	Sym		Sym/Asym	Sym	Sym
Flexibility					Yes
Forward secrecy		Yes	Yes	Yes	Yes
Info. hiding	Yes		Yes		Yes
Keys randomness	Low	Medium	Low	Medium	High
Scalability	Low	Low	Medium	Low	High

6. Conclusions and Future Trends

For continuous data collection and monitoring, a wireless sensor network generally comprises sensor nodes dispersed in areas sensitive to data, such as the health sector. All sensor nodes gather data, which is then transmitted either directly or indirectly to the base station. Due to the nature and variety of applications of WSNs, Security has constantly been a serious problem. In a heterogeneous/hierarchical WSN, for securing connections in all hops a security method has been proposed. This approach provides strong security by attaining confidentiality (RC5), management (Bloom) and randomness (PRNG), in which the information is encrypted/decrypted and authenticated in each stage until it reaches the target node, resulting in increased secrecy of the transmitted message. In addition, this approach has great scalability and flexibility. Furthermore, our proposed method provides high node capture resistance, as the attacker must capture (λ + 1) of cluster heads to compromise the cluster heads' keys. Whereas the capture of a member node has no impact on the other nodes because each member node possesses key information that is unique. However, the sensor's resource is employed in an inequality manner to ensure network balance, resulting in a WSN method that is both efficient and secure. For future directions, we intend to investigate more Bloom filters to support key management. In addition, the accountability requirement will support the robustness of the key management scheme if added to all network devices, which will enhance the security of network device key ownership. Furthermore, we plan to extend the use of WSN within IoT applications to quickly transmit sensor-collected health data anywhere but this will require more attack testing and performance evaluation with more modern methods.

Author Contributions: Contributions to the paper were made by all authors. Conceptualization, R.A.M., N.A.F. and M.A.-Z.; methodology, R.A.M., N.A.F. and M.A.-Z.; software, R.A.M., N.A.F. and M.A.-Z.; validation, M.A.-Z.; formal analysis, M.A.-Z.; investigation, R.A.M. and M.A.-Z.; writing—original draft preparation, R.A.M., N.A.F. and M.A.-Z.; writing—review and editing, M.A.-Z.; project administration, R.A.M. and M.A.-Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

This paper uses the following abbreviations:

Auth-*K*	Authentication key
BS	Base station
CBC-RC5	Cipher block chaining-Rivest cipher 5
CH	Cluster head
HEED	Hybrid energy efficient distributed
ID	Identifier
PRNG	Pseudo-random number generator
SH-*K*	Shared key
Sym/Asym	Symmetric/Asymmetric

References

1. Patil, H.K.; Szygenda, S.A.; Szygenda, S.A. *Security for Wireless Sensor Networks Using Identity-Based Cryptography*; CRC Press: Boca Raton, FL, USA, 2013.
2. Huanan, Z.; Suping, X.; Jiannan, W. Security and application of wireless sensor network. *Procedia Comput. Sci.* **2021**, *183*, 486–492. [CrossRef]
3. Awaad, M.H.; Jebbar, W.A. Study to analyze and compare the LEACH protocol with three methods to improve it and determine the best choice. *J. Comput. Sci. Control. Syst.* **2014**, *7*, 5.
4. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. REISCH: Incorporating lightweight and reliable algorithms into healthcare applications of WSNs. *Appl. Sci.* **2020**, *10*, 2007. [CrossRef]
5. Banerjee, A.; De, S.K.; Majumder, K.; Das, V.; Giri, D.; Shaw, R.N.; Ghosh, A. Construction of effective wireless sensor network for smart communication using modified ant colony optimization technique. In *Advanced Computing and Intelligent Technologies*; Springer: Singapore, 2022; pp. 269–278.
6. Khalaf, O.I.; Romero, C.A.T.; Hassan, S.; Iqbal, M.T. Mitigating hotspot issues in heterogeneous wireless sensor networks. *J. Sens.* **2022**, *2022*, 7909472. [CrossRef]
7. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. RAMHU: A new robust lightweight scheme for mutual users authentication in healthcare applications. *Secur. Commun. Netw.* **2019**, *2019*, 3263902. [CrossRef]
8. Majid, M.; Habib, S.; Javed, A.R.; Rizwan, M.; Srivastava, G.; Gadekallu, T.R.; Lin, J.C.W. Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review. *Sensors* **2022**, *22*, 2087. [CrossRef]
9. Al-Zubaidie, M. Implication of lightweight and robust hash function to support key exchange in health sensor networks. *Symmetry* **2023**, *15*, 152. [CrossRef]
10. Sastry, A.S.; Sulthana, S.; Vagdevi, S. Security threats in wireless sensor networks in each layer. *Int. J. Adv. Netw. Appl.* **2013**, *4*, 1657.
11. Lee, C.-C. Security and privacy in wireless sensor networks: Advances and challenges. *Sensors* **2020**, *20*, 744. [CrossRef]
12. Barati, H. A hierarchical key management method for wireless sensor networks. *Microprocess. Microsyst.* **2022**, *90*, 04489.
13. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. PAX: Using pseudonymization and anonymization to protect patients' identities and data in the healthcare system. *Int. J. Environ. Res. Public Health* **2019**, *16*, 1490. [CrossRef] [PubMed]

14. Al-Zubaidie, M.H.A. Incorporating Security into Electronic Health Records Based Healthcare Systems with Wireless Sensor Networks. Ph.D. Dissertation, University of Southern Queensland, Darling Heights, QLD, Australia, 2020.
15. Ghosal, A.; Conti, M. Key management systems for smart grid advanced metering infrastructure: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2831–2848. [CrossRef]
16. Zhou, H.; Lv, K.; Huang, L.; Ma, X. Security assessment and key management in a quantum network. *arXiv* **2019**, arXiv:1907.08963.
17. Al-Zubaidie, M.; Zhang, Z.; Zhang, J. User authentication into electronic health record based on reliable lightweight algorithms. In *Handbook of Research on Cyber Crime and Information Privacy*; IGI Global: Hershey, PA, USA, 2021; pp. 700–738.
18. Shahzadi, R.; Anwar, S.M.; Qamar, F.; Ali, M.; Rodrigues, J.J. Chaos based enhanced RC5 algorithm for security and integrity of clinical images in remote health monitoring. *IEEE Access* **2019**, *7*, 52858–52870. [CrossRef]
19. Li, L.; Wang, X. A high security dynamic secret key management scheme for wireless sensor networks. In Proceedings of the Third International Symposium on Intelligent Information Technology and Security Informatics, Jinan, China, 2–4 April 2010; pp. 507–510.
20. Iwendi, C.; Allen, A.; Offor, K. Smart security implementation for wireless sensor network nodes. *J. Wirel. Sens. Netw.* **2015**, *1*, 1.
21. Zhang, Y.; Pengfei, J. An efficient and hybrid key management for heterogeneous wireless sensor networks. In Proceedings of the 26th Chinese Control and Decision Conference (2014 CCDC), Changsha, China, 31 May 2014–2 June 2014; pp. 1881–1885.
22. Zhang, X.; Wang, J. An efficient key management scheme in hierarchical wireless sensor networks. In Proceedings of the 2015 International Conference on Computing, Communication and Security (ICCCS), Pointe aux Piments, Mauritius, 4–5 December 2015; pp. 1–7.
23. Qin, D.; Jia, S.; Yang, S.; Wang, E.; Ding, Q. A lightweight authentication and key management scheme for wireless sensor networks. *J. Sens.* **2016**, *2016*, 1547963. [CrossRef]
24. Moara-Nkwe, K.; Shi, Q.; Lee, G.M.; Eiza, M.H. A novel physical layer secure key generation and refreshment scheme for wireless sensor networks. *IEEE Access* **2018**, *6*, 11374–11387. [CrossRef]
25. Chanda, A.; Sadhukhan, P.; Mukherjee, N. Key management for hierarchical wireless sensor networks: A robust scheme. *EAI Endorsed Trans. Internet Things* **2020**, *6*, 23. [CrossRef]
26. Jia, C.; Ding, H.; Zhang, C.; Zhang, X. Design of a dynamic key management plan for intelligent building energy management system based on wireless sensor network and blockchain technology. *Alex. Eng. J.* **2021**, *60*, 337–346. [CrossRef]
27. Ahlawat, P.; Dave, M. An attack resistant key predistribution scheme for wireless sensor networks. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 268–280. [CrossRef]
28. Kumar, V.; Malik, N. Enhancing the connectivity and resiliency of random key pre-distribution schemes for wireless sensor network. *Int. J. Syst. Assur. Eng. Manag.* **2022**, *13*, 92–99. [CrossRef]
29. Tyagi, P.; Kumari, S.; Alzahrani, B.A.; Gupta, A.; Yang, M.H. An enhanced user authentication and key agreement scheme for wireless sensor networks tailored for IoT. *Sensors* **2022**, *22*, 8793. [CrossRef]
30. Liu, J.; Liu, L.; Liu, Z.; Lai, Y.; Qin, H.; Luo, S. WSN node access authentication protocol based on trusted computing. *Simul. Model. Pract. Theory* **2022**, *117*, 102522. [CrossRef]
31. Wang, C.; Wang, D.; Tu, Y.; Xu, G.; Wang, H. Understanding node capture attacks in user authentication schemes for wireless sensor networks. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 507–523. [CrossRef]
32. Ullah, Z. A survey on hybrid, energy efficient and distributed (HEED) based energy efficient clustering protocols for wireless sensor networks. *Wirel. Pers. Commun.* **2020**, *112*, 2685–2713. [CrossRef]
33. Gupta, P.; Sharma, A.K. Clustering-based optimized HEED protocols for WSNs using bacterial foraging optimization and fuzzy logic system. *Soft Comput.* **2019**, *23*, 507–526. [CrossRef]
34. Awaad, M.H.; Jebbar, W.A. Prolong the lifetime of WSN by determining a correlation nodes in the same zone and searching for the best not the closest CH. *Int. J. Mod. Educ. Comput. Sci.* **2014**, *6*, 31. [CrossRef]
35. Mishall Hammed, A. Improve the effectiveness of sensor networks and extend the network lifetime using 2BSs and determination of area of CHs choice. *J. Comput. Sci. Control. Syst.* **2014**, *7*, 15.
36. Anitha, G.; Vijayakumari, V.; Thangavelu, S. A comprehensive study and analysis of LEACH and HEED routing protocols for wireless sensor networks—With suggestion for improvements. *Indones. J. Electr. Eng. Comput. Sci.* **2018**, *9*, 778–783. [CrossRef]
37. Boudhiafi, W.; Ezzedine, T. Optimization of multi-level HEED protocol in wireless sensor networks. In *Communications in Computer and Information Science: International Conference on Applied Informatics*; Springer: Cham, Swidzerland, 2021; pp. 407–418.
38. Jamil, A.S.; Rahma, A.M.S. Image encryption based on multi-level keys on RC5 algorithm. *iJIM* **2022**, *16*, 101.
39. Raj, K.V.; Ankitha, H.; Ankitha, N.G.; Hegde, L.K. Honey encryption based hybrid cryptographic algorithm: A fusion ensuring enhanced security. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; pp. 490–494.
40. Alenezi, M.N.; Alabdulrazzaq, H.; Mohammad, N.Q. Symmetric encryption algorithms: Review and evaluation study. *Int. J. Commun. Netw. Inf. Secur.* **2020**, *12*, 256–272.
41. Sadhya, D.; Sing, S.K. Providing robust security measures to bloom filter based biometric template protection schemes. *Comput. Secur.* **2017**, *67*, 59–72. [CrossRef]
42. Lim, K.; Liu, W.; Wang, X.; Joung, J. SSKM: Scalable and secure key management scheme for group signature based authentication and CRL in VANET. *Electronics* **2019**, *8*, 1330. [CrossRef]
43. Sampangi, R.V.; Sampalli, S. Metamorphic framework for key management and authentication in resource-constrained wireless networks. *Int. J. Netw. Secur.* **2017**, *19*, 430–442.

44. Ghorpade, S.; Zennaro, M.; Chaudhari, B.S. Towards green computing: Intelligent bio-inspired agent for IoT-enabled wireless sensor networks. *Int. J. Sens. Netw.* **2021**, *35*, 121–131. [CrossRef]
45. Mcginthy, J.M.; Michaels, A.J. Further analysis of prng-based key derivation functions. *IEEE Access* **2019**, *7*, 978–995. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.