Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.012806
Article

Tech Science Press

Secure and Energy Efficient Data Transmission Model for WSN

Anuj Kumar Singh[1], Mohammed Alshehri[2,*], Shashi Bhushan[3], Manoj Kumar[3], Osama Alfarraj[4] and Kamal Raj Pardarshani[5]

[1]Krishna Engineering College, Ghaziabad, India
[2]College of Computer and Information Sciences, Majmaah University, Majmaah, 11952, Saudi Arabia
[3]School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India
[4]Computer Science Department, Community College, King Saud University, Riyadh, Saudi Arabia
[5]Department of Mathematics, Bioinformatics and Computer Applications, MANIT, Bhopal, India
[*]Corresponding Author: Mohammed Alshehri. Email: ma.alshehri@mu.edu.sa

Received: 30 August 2020; Accepted: 01 November 2020

Abstract: Wireless sensor networks (WSNs) have been used in numerous delicate checking, observation, and surveillance systems that use sensitive data. When a WSN utilizes various data clustering techniques, data is moved to the cluster head (CH) of the corresponding cluster area. Cluster Head further communicates the information to the sink node. In a WSN, a network owner (NO) does not validate a sensor before connecting to the network, so faulty nodes may likely attach to the network to sense the data and attempt to release the information to unauthorized persons. Further, a malicious node may become a mobile node (MN) equipped to send all of a particular cluster area's perceived data to unauthorized persons. The above-stated problems can be solved by introducing an authentication mechanism into wireless sensor networks. In this mechanism, at whatever point of time a sensor connects with a cluster region, the identity of the sensor must be validated and authenticate to, in turn, validate a mobile node. The perceived data are encrypted and then transmitted through the WSN's transmission medium. However, the encrypted data are prone to flaws, and attackers can gain access to it illegitimately. In this study, a more energy effective and proficient secured model is proposed called a data transmission model. The technical components, including the data access control mechanism and various private key cryptography algorithms, are compared to choose the optimal energy-efficient cryptography algorithm. The proposed model is verified by experiments for encryption and decryption processes using JAVA language with advanced encryption standard (AES), data encryption standard (DES), Triple DES, Rivest Cipher (RC4), and Blowfish algorithms. The encryption time is determined using mathematical equations. The experiment results showed that the Blowfish algorithm was comparatively more energy effective than the other private key cryptography algorithms.

Keywords: Authentication; cluster head; network security; wireless sensor networks

1 Introduction

The information access approach can be considered secure if the user has been authorized first and can only access the source node's desired information (SN). An algorithm that is energy efficient is implemented for both sending and receiving secure information using encryption and decryption processes, respectively [1].

It is initiated with the basic features of wireless sensor networks and issues taking place in practical applications. Motivation towards the present investigation leads to advance features in wireless sensor networks. Initially, the power is generated from the nodes and it is sent to the processing unit. Here the power is passed to the sensing unit, where the sensors collect the information and the output signal is produced.

1.1 Methods to Get Crypto-Based Access Information

The crypto-based admittance control and the job-based admittance control are two sorts of made sure about information access moves toward that have been effectively applied in numerous information access control circumstances [2]. Information access control is a basic portion of the information access measure, and information access control and log management were clearly defined in this process [3]. The Role Based access control mechanism was mainly developed so that all the nodes are assigned with the same functions. Here, the Sensor Protocol for Information Negotiation takes place. It will send large collections of data to the neighbor nodes. It avoids redundant data transmission. By saving the node energy, it will increase the network lifetime. By diffusion, the base station sends the query to the remaining nodes. The transmitted data will send back to the base station to find the optimal route path. The main principle is to determine the buffer length and rate control packets by considering each node will maintain the steady transmission rate and packet arrival time. The principal request energy utilization model is based on this postulation [4]. The energy spent for transmitting information consists of two parts: a) Dissipation caused by hardware and b) Energy required for running the transmitter.

Data security, security mechanisms, and activity patterns play a vital role in data security in practice. The sensors first sense data from a hostile environment and then transmit it via the communication medium to the sink node. Unauthorized users manage to tap the transmitted data in order to misuse it. Another topic to investigate regarding authenticity is the concept of log management. Kurp et al. [4], a secure log management approach was proposed for maintaining user activities regarding data access. This secure log management consists of three layers. The discovery of temporal features and relations of activity patterns (DTFRA) has been used to capture time-related information from data. The applications using this secure activity pattern have been detected as anomalies and marked as unauthorized for data access. In the DTFRA, data activity patterns denote the input, and a set of temporal relations denote the output. In the field of sensors, the input collected as data is the data's identity and timestamp [5]. The output captures the duration of work from the temporal relation. This model also uses the apriori algorithm for finding a secure activity pattern in a hostile environment. This algorithm determines the order of activities and can be used to maintain the log.

1.2 Methods to Manage Logs and Access Control Information

To give a safe access control mechanism and methods to manage logs mechanism, the model shown in Fig. 1 is applied for managing data security [5]. As displayed in Fig. 1, this model consists of three components: (1) User, which can be either a network owner or a nominal user, (2) The CRBAC; (3) Associated log and data. The methodology for working is as follows. Users, who are either the owner or nominal users, can use the data according to their roles [6]. When a user wants to access the information, he has first to send a request to access control and after getting confirmation, only he can access the information. It is the responsibility of CRBAC to authenticate the user based on the user's credentials and permissions, at which point it grants access to retrieve the data. Next, the user sends a data retrieval request to the data component, allowing the user to be authentic. The data component now starts checking whether the user is directly requested for the data or has already requested for the permission of CRBAC and begins maintaining the logs if permitted. If any unauthorized users get detected in managing Logs and access control information, that particular user will become unauthorized.

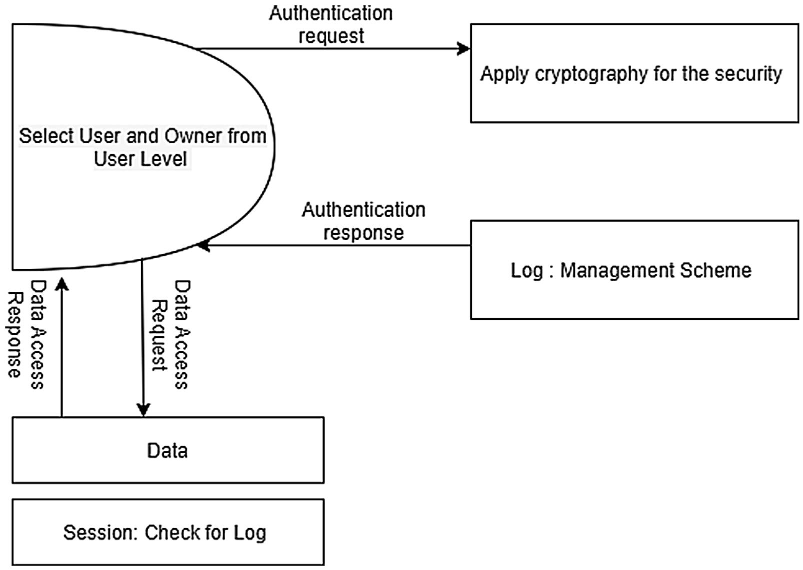

Figure 1: Secure access control and log management

All the User activities in this security-based access control mechanism are mentioned in Tab. 1. These activities are denoted for communication (e.g., 1 to 4), as shown in Fig. 1. This architecture's major points are that the CRBAC secures the access control mechanism and the main thing about this protocol is that it will improve the network's stability [7]. As per our observation, this protocol achieves only 52% of throughput and fairness efficiency is low. Here, in this newly proposed protocol, is a hybrid topology where the overhearing problem is also rectified. The sample period is measured for every time and by varying features; the throughput is also measured for different time periods.

Table 1: Mechanism and strategy: Log methodology and access control

Communication line	Description
1. Request to get authenticate	Make a request to get security as an authentication.
2. Getting response	Receive the response to get authentic and also maintain the log mechanism.
3. Request for the data	Once get a response from step 2 then request for the data access.
4. Getting Data as a response	After getting access to data, now detailed information is also maintained.

2 Relative Comparison on Key Based Security Algorithms

Various key-based security algorithms have been studied and implemented for security practices; one standard algorithm is RC4 algorithm [8]. The secure key-based algorithms are used in encryption and decryption processes to send data from one node to another securely. So, the benefits of using a secure key-based algorithm are that encryption and decryption times are reduced, and the cracking process is also more time-consuming, making the process more effective and efficient.

This works very effectively in WSN, where each node of WSN has fix and less energy, so energy utilization is a crucial factor while selecting a secure key-based algorithm to encrypt data at the sensor nodes. The energy utilization is dependent on the time taken by the protocols to encrypt and decrypt the code. Some of the private key cryptography algorithms discussed in the below table and their structures, key sizes, and round numbers are presented in Tab. 2 [9].

Table 2: Comparative analysis for algorithms

Algorithm	Type of cipher used	Key size	Rounds (in numbers)
Advanced Encryption Standard	Substitution Cipher	64, 128, 256	8, 16, 24
Data Encryption Standard	Feistel Cipher	64	16
T-DES	Feistel Cipher	112, 168	56
Blowfish	Feistel Cipher	32 to 512	16
RC4	Unbalanced Feistel Cipher	92	32

3 Key Based Secure Block Diagram of WSN

The key-based secure block diagram WSN [10] is presented in Fig. 2, where it may be observed that this model comprises Network Owner, Mobile Node, cluster sensor, and authenticator. Here in this safe model, the security validation and encryption process are discussed [11–15]. The function of the Network Owner is to transferred to the secure query to the Mobile Node to read sensor data stored in the Mobile Node. If the Network Owner doubts or feels that the Mobile Node is transferring a malicious data reply, it will validate the same data with the sensor. The Mobile Nodes' primary duty is to accept the Network Owner's query and react to the query using the stored encrypted data and to store the encrypted data transferred by the Cluster Head.

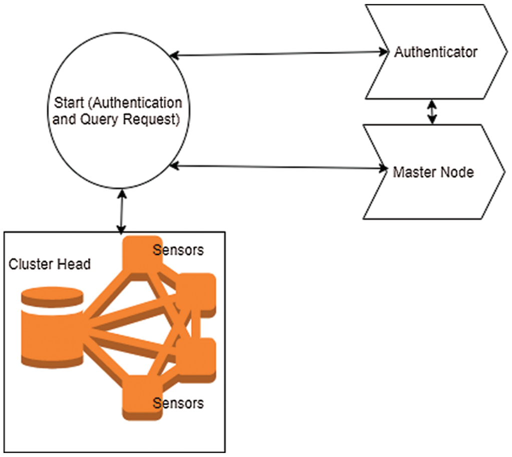

Figure 2: Secure model architecture for the WSN using private keys

A feedback control loop is designed to determine the total rate of control packets. The light-weight buffer technique will produce only 50% of throughput. To improve this, a globalized probabilistic dropping packets algorithm is designed. In this technique, the controller is designed and an error message is sent to every node. If an error is detected, it starts recovering the packets or data. This leads to improving throughput and network performance. Then the validator generates specific keys with the help of a specially designed algorithm and those keys are passed on sink nodes as well as Mobile nodes when they will join the network

The processing component of this model architecture is explained in Fig. 2. First, the validator produces the encryption keys or decryption keys for each sensor on regular intervals. The sensor stores the manipulated or converted data till the next time interval arises. For instance, if the sensor senses the recently changed data, then the same data will be there for the next time interval. In the next phase, the Mobile Node accepts the data in the encrypted form and saves the sensor information identification number and time when the sensor transmits the data [16]. The validator saves and stores each key for each sensor in the database during the Network Owner decryption process. The Network Owner sends the following query to the

Mobile Node: "Send the data sensed by sensor s1 at Cluster Head 1 at time epoch t1."

Then, the Mobile Node receives the query and sends the request to the authenticator as follows:

"Send the decryption key k1 assigned for s1 of Cluster Head1 at t1."

3.1 Blowfish Algorithm

The Blowfish algorithm has shorter encryption and decryption times than similar algorithms. The Blowfish algorithm's primary advantage is that it has significantly less encryption and decryption times than various other similar algorithms, and much longer time is required for unauthorized users to break it down. That is why it is preferably suggested to use the Blowfish algorithm, and it can act as a secure key-based algorithm for safe data transmission in WSNs. One of the parameters that give more preference to the Blowfish algorithm is that it can run at a memory size of less than 5 KB. The Blowfish algorithm is simple to implement and for better security; it uses the additional factor of adding XOR operation and lookup table in each round. The other competitive block ciphers, such as Advanced Encryption Standard, DES, and Triple DES, can be used to require more complex mathematical operations for data transfer. Last, we can say it is easier and safer than other similar algorithms where the size for changing from 32 bits to 448 bits [12,17]. The Blowfish algorithm uses 64 bits of data and the Feistel network for 16 rounds, as shown in Fig. 3.

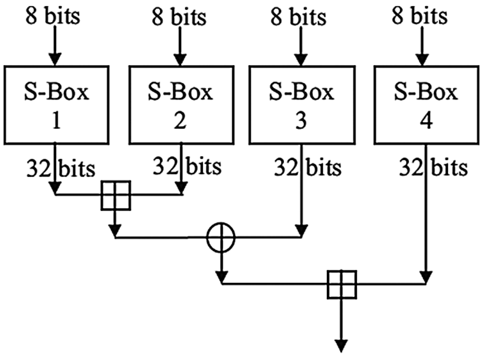

Figure 3: Round function in the Blowfish algorithm

The Blowfish algorithm processes 64 bits of data per block using the key size of 32–448 bits in 16 rounds in the Feistel network. The Blowfish algorithm has been used for various data security purposes, including secure data transmission and secure storage, and it has recently been applied to the Internet of Things (IoT) [18–20]. There are many examples where the Blowfish algorithm was used for encryption and decryption processes, and some of them are mentioned in the following. The Blowfish block cipher algorithm was used for secure data transmission in the IoT [19]. Suresh et al. [21], the Blowfish algorithm round function was studied, and its encryption time and throughput were compared with those of the other block cipher algorithms, including the DES, AES, RC4, and Triple-DES. The

Blowfish algorithm outperformed the other algorithms. With a very large-scale implementation, the effect of the Blowfish algorithm in secure communication was discussed in Suresh et al. [21]. Suresh et al. [21] compared the AES and Blowfish block cipher algorithms. The results showed that performance parameters, including the encryption speed, were higher while the CPU utilization and battery power were lower for the Blowfish algorithm than for the AES block cipher algorithm.

The block cipher algorithms were used for multimedia data security in Saraereh et al. [14]. The encryption times of the RSA, DES, Blowfish, and RC4 algorithms were compared in Kaswan et al. [19]. The results showed that the Blowfish algorithm performed the best in multimedia data encryption and decryption processes. The secure WSN network was implemented using a Blowfish block cipher for patient data transmission in Suresh et al. [21]. The patient's information monitoring wearable sensor was attached to the needy patient in Kaswan et al. [19]. The data were captured from the human body and converted into ciphered data using the Blowfish algorithm and sent via a wireless medium. The secured encrypted patient data were decrypted at the doctor's side, and the patient's health information was monitored. The advantage of this method is that the patient data are accessed securely, and the doctor can monitor the patient from any place and any time. Another interesting application of data security is video data security. The most challenging task for providing security to video data, as encryption and decryption, is choosing an appropriate cryptography algorithm. The encrypted H.264 video transmission using the Blowfish algorithm was proposed for data security in Suresh et al. [21]. There are many real-time applications, which are online as movies and as a camera and sensor-based WSN.

4 Experimental Results

The experiment was conducted using MATLAB to cluster the sensor nodes. The Java programming language was used to implement the Blowfish algorithm with an electronic code block (ECB) cipher-block mode of operation, which was then compared with Triple DES, DES, and AES security algorithms. The system used in this experiment was the 64-bit microprocessor with a 3.06 GHz I_3 processor with 4 GB RAM. The data size for the encryption process for all block cipher algorithms was set to 20 KB, 40 KB, 60 KB, 80 KB, and 100 KB, successively, and the corresponding encryption and decryption times were saved.

4.1 Encryption Time Comparison of Block Ciphers

For the data size of 40 KB, the Blowfish algorithm encryption time was 2 s, and for DES, AES, RC4, and Triple DES, the encryption times were 3 s, 5 s, 8 s, and 10 s, respectively. The Blowfish encryption times for the data sizes of 60 KB, 80 KB, and 100 KB were also considerably less than other block ciphers' encryption times. The encryption time comparison of the Blowfish, DES, AES, RC4, and Triple DES block cipher algorithms for the data sizes of 20 KB, 40 KB, 60 KB, 80 KB, and 100 KB, is given in Tab. 3.

Table 3: Encryption time comparison

Input size	Execution time for encryption				
	3DES	RC4	AES	DES	BF
20 KB	5	4	3	1	1
40 KB	10	8	5	3	2
60 KB	18	12	8	5	4
80 KB	20	16	11	8	6
100 KB	25	20	14	10	8

The mathematical model used for energy-efficient secure data transmission in the experiment is given by Eqs. (1)–(3), which define the encryption, encryption time, and speed, respectively.

$$Encryption \ = Enc\_X(Data, \ Key) \tag{1}$$

where *Enc_X* denotes the security algorithm type of, *Data* denotes the data size in bytes, and *Key* denotes the key data in bytes.

$$Enc\_Time \ = \frac{Data\_Size}{Speed} \tag{2}$$

$$\text{Speed} = \frac{\text{Cycle per Second}}{\text{Cycle per Byte}} \tag{3}$$

In Eq. (2), *Enc_Time* denotes the encryption time measured in s, *Data_size* denotes the data size, and *Speed* denotes the ratio of the cycles-per-second to the cycles-per-bytes.

The encryption times of different block cipher algorithms for different data sizes are presented in Fig. 4. The Blowfish algorithm's encryption time was shorter than those of the DES, AES, RC4, and Triple DES algorithms for all the data sizes.

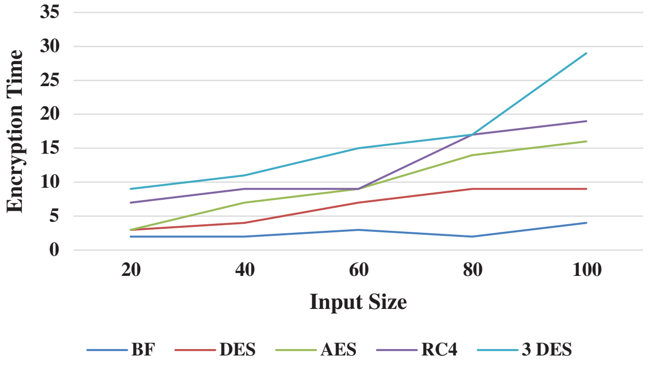

Figure 4: Encryption time comparisons of block cipher algorithms

The decryption times of different are compared in Fig. 5, where it can be seen that the Blowfish algorithm had a shorter decryption time than the DES, AES, RC4, and Triple DES algorithms for all data sizes shows in Tab. 4.

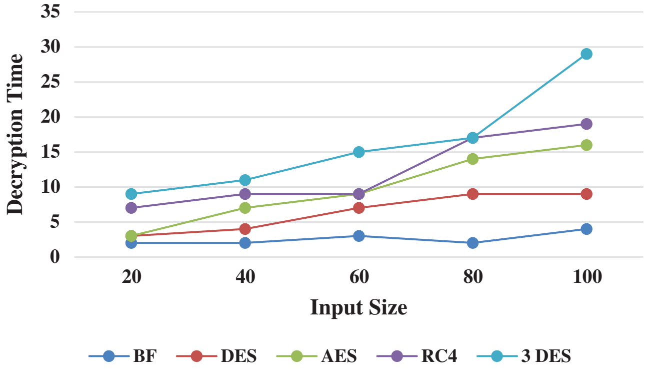

Figure 5: Decryption time comparisons of block cipher algorithms

Table 4: Decryption time comparison

Input size	Execution time for encryption				
	3DES	RC4	AES	DES	BF
20 KB	4	2	3	2	2
40 KB	9	9	7	4	3
60 KB	16	14	9	7	3
80 KB	19	17	9	9	7
100 KB	29	17	15	11	9

The time required to encrypt data of various using the Blowfish algorithm was comparatively shorter than those of the DES, AES, RC4, and Triple DES algorithms. Therefore, the Blowfish algorithm was identified as a better solution for the proposed secured data transmission model. The experiment confirms that the Blowfish algorithm can perform complex encryption operations, such as key whitening, applying S-box, $EXOR$ with the F function, and output swapping, in less time than the other algorithms. Thus, the Blowfish algorithm represents an energy-efficient algorithm for secure data transmission.

5 Conclusion

This paper presents an energy-efficient secure data transmission model and its components. The technical components, including the data access control mechanism and various private key cryptography algorithms, are compared to choose the optimal energy-efficient cryptography algorithm. The experiment was conducted using Java programming language, and it conducted encryption and decryption processes. In the experiments, the performances of the AES, DES, Triple DES, RC4, and Blowfish algorithms were compared in terms of the encryption and decryption times, which were calculated using the mathematical equations. The experiment results showed that the Blowfish algorithm was comparatively more energy efficient than the other private key cryptography algorithms. The Blowfish algorithm's strengths and merits were intense, and its complex mathematical functions took a relatively short time to encrypt and decrypt the data. The Blowfish algorithm performed better than all the other comparison algorithms for all the data sizes used in the experiment. Thus, the Blowfish algorithm is used for secure data transmission in the proposed energy-efficient secured data transmission model. In future, more cryptographic algorithms can be implemented for third party secure communications so that efficiency in multiway can be improved.

Acknowledgement: The authors would like to express their heartfelt thanks to the editors and anonymous referees for their most valuable comments and constructive suggestions, which led to significant improvements in the earlier version of this manuscript.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific research at Majmaah University for funding this work under Project No. (RGP-2019-26).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. Li, C. Wang, Z. Yang, L. Yan and S. Han, "Energy-efficient and secure transmission scheme based on chaotic compressive sensing in underwater wireless sensor networks," *Digital Signal Processing*, vol. 81, pp. 129–137, 2018.

[2] W. Y. Alghamdi, H. Wu and S. Salil, "Reliable and secure end-to-end data aggregation using secret sharing in WSNs," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, CA, pp. 1–6, 2017.

[3] R. Pawar and D. R. Kalbande, "Elliptical curve cryptography based access control solution for IoT based WSN," *Innovative Data Communication Technologies and Application. Lecture Notes on Data Engineering and Communications Technologies*, vol. 46. Cham.: Springer, pp. 742–749, 2019.

[4] T. Kurp, R. X. Gao and S. Sah, "An adaptive sampling scheme for improved energy utilization in wireless sensor networks," in *2010 IEEE Instrumentation & Measurement Technology Conf. Proc.*, Austin, TX, pp. 93–98, 2010.

[5] S. Yu, K. Ren and W. Lou, "FDAC: Toward fine-grained distributed data access control in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 4, pp. 673–686, 2011.

[6] J. Duan, C. Deyun and Hongke, "TC-BAC: A trust and centrality degree-based access control model in wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2675–2692, 2013.

[7] F. Ma, M. Yong and M. Fuchun and Ding, "The fine-grained security access control of spatial data," in *2010 18th Int. Conf. on Geoinformatics*, Beijing, pp. 1–4, 2010.

[8] F. Barbosa, H. Arthur and F. L. Mello, "Machine learning applied to the recognition of cryptographic algorithms used for multimedia encryption," *IEEE Latin America Transactions*, vol. 15, no. 7, pp. 1301–1305, 2017.

[9] J. Kim and S. Nepal, "A cryptographically enforced access control with a flexible user revocation on untrusted cloud storage," *Data Science and Engineering*, vol. 1, no. 3, pp. 149–160, 2016.

[10] M. K. Venkateswarlu, A. Kandasamy and K. Chandrasekaran, "An energy-efficient clustering algorithm for edge-based wireless sensor networks," *Procedia Computer Science*, vol. 89, pp. 7–16, 2016.

[11] M. Upmanyu, A. M. Namboodiri, K. Srinath and C. V. Jawahar, "Blind authentication: A secure crypto-biometric verification protocol," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 255–268, 2010.

[12] Mu Arvindhan and Anand, "The modern way for virtual machine placement and scalable technique for reduction of carbon in green combined cloud datacenter," *Green Computing in Smart Cities: Simulation and Techniques*, Springer International Publishing, 2020.

[13] S. Wei, J. Wang, R. Yin and J. Yuan, "Trade-off between security and performance in block ciphered systems with erroneous ciphertexts," *IEEE Transactions on Information Forensics & Security*, vol. 8, no. 4, pp. 636–645, 2013.

[14] O. A. Saraereh, I. Khan and B. M. Lee, "An efficient neighbor discovery scheme for mobile WSN," *IEEE Access*, vol. 7, pp. 4843–4855, 2019.

[15] T. Niknam, A. Kavousifard and J. Aghaei, "Scenario-based multiobjective distribution feeder reconfiguration considering wind power using adaptive modified particle swarm optimisation," *IET Renewable Power Generation*, vol. 6, no. 4, pp. 236–247, 2012.

[16] Q. Yang, X. Zhu, H. Fu and X. Che, "Survey of security technologies on wireless sensor networks," *Journal of Sensors*, vol. 2015, pp. 1–9, 2015.

[17] Y. S. Khiabani, S. Wei, J. Yuan and J. Wang, "Enhancement of secrecy of block ciphered systems by deliberate noise," *IEEE Transactions on Information Forensics & Security*, vol. 7, no. 5, pp. 1604–1613, 2012.

[18] A. Ostad, H. Arshad, M. Nikooghadam and D. Abbasinezhad-Mood, "Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme," *Future Generation Computer Systems*, vol. 100, pp. 882–892, 2019.

[19] A. Kaswan, K. Nitesh and P. K. Jana, "Energy efficient path selection for mobile sink and data gathering in wireless sensor networks," *AEU-International Journal of Electronics and Communications*, vol. 73, pp. 110–118, 2017.

[20] S. Bhushan, A. K. Singh and S. Vij, "Comparative study and analysis of wireless mesh networks on AODV and DSR," in *4th Int. Conf. on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Ghaziabad, India, pp. 1–6, 2019.

[21] M. Suresh and M. Neema, "Hardware implementation of blowfish algorithm for the secure data transmission in internet of things," *Procedia Technology*, vol. 25, pp. 248–255, 2016.