Journal of Network and Computer Applications 202 (2022) 103353

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

SEEDGT: Secure and energy efficient data gathering technique for IoT applications based WSNs

Ahmed Salim [a,e], Walid Osamy [b,c,*], Ahmed Aziz [c,f], Ahmed M. Khedr [d,e]

[a] Department of Computer Science, College of Science and Arts, Al-methnab, Qassim University, P.O. Box 931, Buridah 51931, Al-mithnab, Kingdom of Saudi Arabia
[b] Department of Applied Natural Science, Applied College in Unaizah, Qassim University, Unaizah, Qassim, Kingdom of Saudi Arabia
[c] Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt
[d] Computer Science Department, University of Sharjah, Sharjah, 27272, United Arab Emirates
[e] Faculty of Science, Zagazig University, Zagazig, Egypt
[f] Tashkent State university of Economics, Tashkent, Uzbekistan

ARTICLE INFO

Keywords:
Cyber security
Cluster head selection
Compressive sensing
Internet of Things
Network lifetime
Public key
Trust management
Wireless sensor networks

ABSTRACT

Internet of Things based Wireless Sensor Networks (IoT-WSNs) are widely employed in a variety of applications, including military, health-care, and industrial monitoring. Security parameters are essential to protect the network from various security threats and attacks because these applications handle sensitive information in potentially hostile and unattended environments. This paper aims to achieve the security parameters for IoT applications based WSNs by proposing a secure and energy aware data gathering technique, called SEEDGT, which integrates between trust, public key algorithm, and Compressive Sensing (CS) methods towards achieving security with fair energy load balance in IoT-WSN. The proposed SEEDGT technique has three phases, namely Cluster formation, network operation and reconfiguration phase. During the cluster formation phase, energy-aware and trust based methods are applied for the creation of clusters and cluster head selection. The network operation phase aims to achieve security by using the public key algorithm to encrypt network data during the data gathering process. Moreover, in this phase, CS strategy is employed to reduce the original data size, which leads to a reduction in energy usage. Finally, the changes that could occur during network operations is considered through the reconfiguration phase. The simulation studies illustrate that SEEDGT is effective in achieving better performance than other baseline approaches.

1. Introduction

The most essential aspect of the IoT model is the Wireless Sensor Networks (WSNs) because they form the efficient networks for observing and tracking various environmental phenomena. The usage of wireless sensors and IoT innovations in various applications (e.g., e-health, transportation, etc.) is one of the most exciting market segments in the future (Afsar Mehdi et al., 2014; Osamy and Khedr, 2020; Osamy et al., 2019a,b; Aziz et al., 2019; Khedr, 2105). IoT sensor nodes have the main task of detecting application specific data and then transmitting it to the sink or Base Station (BS). One of the significant factors affecting the energy usage of IoT nodes is the data transmission process. In addition, wireless transmission of data using insecure channels make it very easy to eavesdrop. Security and energy efficiency are therefore the keys to the growth of IoT. However, developing security aware technique that takes into consideration the energy parameters for IoT network is not an easy task.

Two groups of encryption algorithms namely symmetrical and asymmetrical key algorithms are commonly used. Data encryption algorithms dependent on symmetric (private) keys such as DES and AES (Karl and Willig, 2007) do not require complex computation and storage. However, these algorithms require a key exchange technique between the sender and the receiver. Otherwise, the keys have to be saved in advance. So when installed in outdoor locations, sensors could be vulnerable to hacking (Khediri et al., 2014; Singh et al., 2017). On the other hand, asymmetric (public) key-based data encryption algorithms such as RSA and ECC (Dhillon and Kalra, 2016) can accomplish an increased level of security. Public-key algorithms are used in IoT-based WSNs in two scenarios: First, each sensor node generates its own public and private key to encrypt and decrypt data, which increases

* Corresponding author at: Computer Science Department, Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt.
E-mail addresses: a.salem@qu.edu.sa, a.salim@zu.edu.eg (A. Salim), w.elsherif@qu.edu.sa, walid.osamy@fci.bu.edu.eg (W. Osamy), ahmed.aziz@fci.bu.edu.eg, a.mohamed@tsue.uz (A. Aziz), akhedr@sharjah.ac.ae, amkhedr@zu.edu.eg (A.M. Khedr).

https://doi.org/10.1016/j.jnca.2022.103353
Received 24 May 2021; Received in revised form 21 November 2021; Accepted 20 February 2022
Available online 12 March 2022

computation and complexity, and reduces network life. Thus, the first scenario cannot be applied to the IoT network. Second, BS generates the keys and transmits the public key to the entire network to encrypt sensor data (Gulen et al., 2019). The encrypted data is then transferred by the sensors to the cluster head (CH) for data aggregation. The problem with this scenario is that cluster heads cannot aggregate sensor data without decrypting the data, i.e. they need a private key, which leads to an increase in data traffic and communication costs. Moreover, nodes have to sacrifice their privacy because traditional encryption systems cannot operate on encrypted data without a prior decryption. For that reasons, it is required to ensure confidentiality and at the same time, perform aggregation of data, i.e., we must be able to aggregate encrypted data.

Homomorphic encryption is an encryption scheme that enables to perform mathematical operations, namely addition and multiplication, on encrypted texts and obtain an encrypted result (as same as the result of mathematical operations on plain text). Homomorphic encryption permits to perform different operations in an untrusted environment without revealing sensitive data for any operation (Rivest et al., 1978; Gentry et al., 2013; Paillier, 1999).

In the proposed technique, a homomorphic encryption scheme (Paillier, 1999) is employed to allow aggregation of encrypted data (in other words, aggregating two data values and then encrypting the result will be equivalent to encrypting each value separately). However, the data size has direct impact on increasing the communication load. Therefore, it is required to compress data before encryption to reduce communication loads. Compressive Sensing (CS) is considered one of the most efficient scheme data gathering method in order to reduce the data size during the transmission process which leads to prolong the network lifetime. In CS, if the signal is sparse either naturally or using a domain transformer, it can be effectively sampled at the rate less than the Nyquist theory. According to CS theory, the raw data x is transformed to the compressed data y by matrix Φ. The proposed technique integrates CS with a homomorphic encryption scheme to effectively achieve energy efficacy as well as security.

Encryption can protect against external threats rather than internal attacks. Meanwhile, trust management mechanisms are one of the most powerful ways to identify and safeguard from internal attacks, where trust is defined based on the node's trust metric or trust ranking as the degree of belief that a node may have on another node in the network (Fang et al., 2020).

Trust management has been accepted as one of the most appropriate approaches for identifying the faulty nodes and malicious nodes in the WSNs. It improves robustness of the network, and ensures the secure delivery of data along with secure resource sharing (Dhulipala and Karthik, 2017; Fang et al., 2020; Yan et al., 2014).

In order to clarify the importance of trust management, consider the example: if a malicious node is selected as the CH, the system performance would be highly affected as all member nodes rely on the CHs for packet forwarding towards sink. Moreover, the data integrity would be affected by malicious nodes. Hence, the selection of a reliable CH with adequate residual energy is crucial for the overall performance of the network. Therefore, to achieve improved security in data gathering, we adopt a trust management method integrated with a clustering method that is aware of energy and the trustworthiness of the nodes.

In summary, our aim is to design a technique called SEEDGT that is energy saving as well as capable of ensuring the security needs, for efficient data gathering in WSNs. Our contributions are summarized below:

1. The proposed technique utilizes trust management approaches for CH selection in order to prevent the compromised nodes from accessing the privileged information.
2. The proposed technique employs the Homomorphic encryption scheme to encrypt the network data, which also ensure confidentiality for IoT-WSNs.
3. In contrast to other public key algorithms that require cluster data decryption during aggregation process, the proposed technique eliminates the need of decrypting the data by utilizing the Homomorphic encryption scheme — aggregation property during the cluster data gathering.
4. The proposed technique utilizes advantages of CS in achieving load balancing, by proposing modified plain CS method which addresses the limitations of traditional plain CS method in order to achieve the best adoption with Homomorphic algorithm. That is why, the proposed technique achieves strong security performance and load balancing through the use of a public key algorithm as well as data vector size reduction through the use of the proposed modified CS method.
5. Due to the energy constraints of the sensor node, our technique shifts all complex computing, keys generation and data decryption operations to the BS side that has no energy problem.

The remainder of the paper is arranged accordingly: In Section 2, we discuss the background study. Related works are briefly reviewed in Section 3. The proposed secure and energy efficient data gathering technique (SEEDGT) is provided in Section 4. Section 5 discusses the details of simulation of the proposed technique as well as comparison with current methods. The paper's conclusion is provided in Section 6.

2. Background study

In this section we provide the background study on Compressive Sensing and Homomorphic Encryption scheme.

2.1. Compressive Sensing (CS)

The general architecture for CS is defined as: N sensor nodes are deployed for sensing data from an environment and then sending it to BS. Let matrix $x(x \in R^{N\times 1})$ reflect the reading of sensors, where each row x denotes a sensor node's data. The x matrix has a sparse representation in the Ψ transformer domain where Ψ is a $N \times N$ transform matrix, that is, $x = \Psi k$, and k is the coefficients vector. To put it in another way, x is S-sparse if it has S values as non-zero & $N - k$ zeros. As per the CS model, if M_s is the size of measurement and S is the sparsity level, then BS needs only $M_s \geq S$ log (N/S) to reproduce the actual node's reading x from the measurements of CS y, $y = \Psi x$, $y \in R^{(M_s\times 1)}$ and Φ is $M_s \times N$ ($M_s \gg N$) random CS matrix (Gaussian, Bernoulli, etc.).

2.2. Homomorphic encryption scheme

Public key cryptography (or asymmetric encryption) has been introduced in Diffie and Hellman (1976). These types of algorithms are designed to solve the key distribution problems in private key algorithms in which the sender and the receiver must agree and share the same key in order to encrypt and decrypt the messages between them through insecure channels. The main idea of public key algorithms is that each communication side generates two pair of keys: private and public, the public one will be shared to everyone in the chat session. So, anyone who wants to send a message can encrypt his message using the public key. The receiver will use his private key to decrypt the cipher text. Hence, public key encryption proves to be a method for encrypting data that only the owner of the private key can decrypt. The public key, which is used to encrypt the data, is accessible to anyone. Relying on that, several public key schemes have been proposed (Rivest et al., 1978; Abdallah et al., 2015), and this paper focuses on one of them, which is known as Homomorphic encryption schemes (Gentry et al., 2013).

Homomorphic encryption schemes allow IoT devices to transfer their data securely anytime to the BS by using the public key concept. Moreover, it has additional useful properties of allowing functions and

operations to be performed over encrypted data, which means that the data is never un-encrypted in an untrusted environment. Based on Homomorphic encryption schemes, Paillier cryptosystem (Paillier, 1999) has been proposed. The Paillier cryptosystem is based on the computationally intensive problem of computing nth residue classes. Upon decryption, the essence of the algorithm allows for homomorphic addition processes to generate the current result. The key generation, encryption equation and decryption equation for Paillier Cryptosystem are given in Algorithms 1, 2 and 3 respectively. Provided a message m and a public key (n, g), the encryption method in the Paillier Cryptosystem meets the following homomorphic property.

$$D(E(m_1) \times E(m_2)) = m_1 + m_2 \tag{1}$$

we can prove it as follows:

$$\begin{aligned} c &= c_1 \times c_2 \mod n^2 \\ &= g^{m_1} \times r_1^n \times g^{m_2} \times r_2^n \mod n^2 \\ &= g^{m_1+m_2} \times r_1^n \times r_2^n \mod n^2 \\ &= g^{m_1+m_2} \times r^n \mod n^2 \\ &\therefore D(c) = m_1 + m_2 \end{aligned}$$

Since the values r_1 and r_2 are random, they can be merged to create another random value r.

Algorithm 1 KEY GENERATION ALGORITHM

1: Choose p and q large prime numbers such that $gcd(pq, (p-1)(q-1)) = 1$
2: Determine $n = pq$
3: Compute $\lambda = lcm(p-1, q-1)$
4: Choose a random integer r, where $g \in Z^*_{n^2}$
5: Specify $L(x) = \frac{x-1}{n}$
6: Ensure that n divides the order of r.
7: $u = (L(r^\lambda \mod n^2))^{-1}) \mod n$
8: Private Key=(λ, u)
9: Public Key=(n, r)

Algorithm 2 ENCRYPTION ALGORITHM

1: ***Input:*** A message m to be encrypted, where $m \in Z_n$
2: Pick a random integer r, where $r \in Z^*_{n^2}$
3: Compute $C = g^m \times r^n \mod n^2$
4: ***Output***:Cipher-Tex C

Algorithm 3 DECRYPTION ALGORITHM

1: ***Input:*** A message c to be decrypted, where $c \in Z_n$
2: Compute $m = L(c^\lambda \mod n^2) \times u \mod n$
3: ***Output***:Plain-Tex m

3. Related works

In this section, we present the related works in the field of data gathering in IoT-based WSNs that utilize approaches including: compressive sensing, trust management and secure data collection schemes. Our methodology for selecting literature content stems from our goal of providing a secure and energy-aware data gathering technique that integrates trust, public key algorithm, and Compressive Sensing (CS) methods in order to achieve security while maintaining a fair energy load balance in IoT-WSNs. We split this section into three subsections, where we discuss the related work in each of the above mentioned contexts.

3.1. Compressive Sensing (CS) based approaches

In Yu et al. (2020), a Compressed Data Gathering method based on Hierarchically Diffused Connected Dominating Sets (HDCDS) is proposed with the aim of balancing node distribution. The Connected Dominating Set (CDS) is constructed by HDCDS using the Maximal Independent Set (MIS), and each node in CDS defines a cluster. The CH reveals to its neighbor nodes that it has become the CH by sending a message containing its own residual energy.

The work in Tirani et al. (2020) offers an approach for data collection that takes into account the CS scheme, sink mobility, and a cluster-based routing technique. In the first stage, a clustering method was implemented and nodes are organized into clusters, while in the second stage, one or more mobile sinks pass across the network and acquire aggregated data from the CHs.

The work in Wang et al. (2019b) adopts a Compressed Data Gathering (CDG) method based on an embedding mechanism in which a privacy preserving method of data gathering is applied through certain nodes that are assigned by the sink to embed the blindness factors and the remaining nodes make use of the confusion factors.

In Pachareney and Gupta (2019), hexagonal clustering of WSN is adopted and a CH is assigned for each hexagon. Each CH is responsible for collecting data from its respective sub-CHs, executing CS process, and transmitting compressed samples to the BS.

In Wang et al. (2019a), another cluster-based routing strategy is introduced. It uses the benefits of CS to provide energy-efficient operation by leveraging spatial–temporal correlation. The authors of ST-CDGA (Zhang et al., 2019) incorporated Kronecker CS (KCS) with cluster scheme to exploit spatiotemporal correlations in simultaneous manner. Each CH creates a sparse sub-measurement matrix from the gathered data in order to avoid sensing nodes from having trouble in communicating their readings effectively.

A mobile intelligent computing-based method of data gathering using CS for IoT networks (MIC-CSDG) is developed to improve data recovery performance, according to Sun et al. (2019). The WSN has been split into different regions. An individual sampling and measuring procedure is carried out in each region. MIC-CSDG employs mobile intelligent computation to determine the proportional (multi-hop) relationship between the nodes.

Lv et al. (2019) presented a data gathering framework for sequential temperature data. To improve temperature measurement accuracy, the method integrates the measurement matrix with the sparsifying basis. The cluster scheme is built by employing CS, which enhances energy efficiency by leveraging spatiotemporal relationship.

The work in Aziz et al. (2021, 2020b) implemented a chain-routing strategy and applied CS-based data collection for IoT-WSNs. The benefits of CS in minimizing the data-size along with the efficient routing of gathered data using chain-based scheme enhanced the system performance.

A data gathering scheme based on Compressive Sparse Sampling (CSS) approach is used in Xu et al. (2019). At the same time, a fuzzy oriented compressive data collection method (Ghaderi et al., 2020) that combines the benefits of data aggregation using CDG (Chong et al., 2009) and data routing using geographic adaptive fidelity techniques to enhance the data collection performance.

A multi-hop approach of data transmission by adopting a cluster-based aggregation framework that utilizes hybrid CS (EMCA-CS) is proposed by Aziz et al. (2020a). EMCA-CS is demonstrated to be efficient for IoT-based heterogeneous WSNs and the results indicate a remarkable increase in network lifespan. Moreover, the data restoration error is significantly reduced.

3.2. Trust management schemes in WSNs

A lightweight group-based scheme for trust management in WSN is introduced (GTMS) in Chatterjee et al. (2002). It employs a hybrid trust management technique to reduce the burden on resource constrained sensor nodes.

A reliable lightweight trust management method (LDTS) eliminating the feedback among cluster nodes in order to reduce the communication overhead is introduced in Smaragdakis et al. (2004). Trusted-based Dynamic Slicing Mechanism (TDSM) is proposed for data aggregation

in WSN by authors of Zhang et al. (2020). TDSM can successfully secure the data integrity and mitigate malicious attacks. Also, the approach decreases traffic and increases lifespan of network.

The algorithm presented in Ramalingam and Audithan (2014) establishes trust between nodes within one-hop distance based on recommendation from neighbor nodes. A design based on node ID, trust value and attributes set is used to represent nodes (Kodali and Soratkal, 2015). If the chance of packet forwarding and encryption rate increases, so does the trust value. A malicious node can be identified by examining its attributes and then assigning a trust value to it.

Narayan et al. (2019) avoids appointing a compromised or malicious node as a CH by selecting a reliable CH. Energy aware and trust based CH election for WSNs (EAT) is suggested in Wei and Yu (2018); this method relies on an efficient and distributed model of trust management and takes residual energy into account during the process of CH election.

Another technique in Narayan and Daniel (2021) is to determine the threshold value for CH election according to the residual energy and the distance to BS, and then use a data fusion process on the basis of a trust factor to obtain reliable data.

The work in Mishra et al. (2019) suggested a technique for selecting CH based on a trust attribute. The data fusion and trust attributes are used in the system to prevent needless transmission and achieve a higher packet transfer ratio.

Salehi and Karimian (2017) uses the neighbors' trust values and shares data with neighbor nodes based on these values. The node whose trust is higher than a threshold value will be a candidate for CH in each cluster. Finally, among the candidate nodes, a node with the most trusted neighbors and a suitable level of energy is chosen as the CH using fuzzy logic.

Another trust based scheme for energy aware and secure routing called EATSRA presented in Selvi et al. (2019) utilized the trust metrics that includes RSSI, residual energy and packet reception ratio to monitor the behavior of nodes and solve the security challenges. It is designed and implemented in WSN to provide optimal and reliable routing. Trust scores are used for intruder detection in WSNs more effectively, and a decision tree-based routing mechanism is used to choose the best safe route. Furthermore, spatiotemporal parameters are utilized to make more accurate routing decisions.

In Fang et al. (2021), a trust management-based LEACH (LEACH-TM) is introduced. It used a number of dynamic decision CH nodes, residual energy, and the density of neighbor nodes to constrain the cluster size and enhance the energy efficiency. In addition, for secure selection of CH, LEACH-TM employs a trust management scheme based on BETA distribution.

A hierarchical clustering scheme based on trust proposed in Gaber et al. (2018) presents a bio-inspired trust-based method of CH selection. It employs the Bat Optimization Algorithm (BOA) with residual energy, trust value, and the neighbors count serving as CH selection criteria. BOA is applied to find a group of nodes that span any node in the network and at the same time, have trust value and residual energy greater than or equal to a user-defined threshold.

3.3. Secure data gathering schemes in WSNs

In the context of secure data gathering in WSNs, ElGamal based sparse compressive data gathering (ESCDG) (Xiaohan et al., 2019) is proposed with the objective to improve the performance of CDG by utilizing the sparsity of the sensing matrix. ESCDG combines ElGamal encryption algorithm and sparse random matrix-based CS technology for secure data collection and efficient resource utilization in WSN.

An adaptable secure CS-based data acquisition scheme for distributed WSN is proposed in Liu et al. (2019). However, according to this scheme, each node performs both encryption and decryption, but these are computationally expensive tasks for resource constrained nodes.

A secure data acquisition system based on CS was developed in Zhang et al. (2018a). The method improves the security and provides better load balancing among nodes. The actual data is successfully recovered even from the noise and analyzed at the BS. The scheme enhances data protection while reducing communication effort. However, because of the use of symmetric encryption key, it has a drawback in terms of key storage space and management.

Fuzzy-based Secure Data Collection (FSDGA) (Samydurai et al., 2020) approach is introduced on the basis of slot-based planning and an encryption scheme with asymmetric keys. Cluster formation and CH election is performed using a voting method. Then, to reduce the vulnerability of attacks, routes are discovered using an authentication metric in terms of specific key credentials. Furthermore, the Mamdani Fuzzy Decision Technique is used in conjunction with a data gathering algorithm to increase the data delivery performance.

Recently, the homomorphic encryption technique is utilized. It allows each node to encrypt the sensed data that only the BS can decrypt.

Mobile agents (MAs) are employed for data collection in Kumar et al. (2015). In Kumar et al. (2015), the node encrypts its data based on homomorphic encryption method and keeps its data secured during aggregation processes by MAs. The MA executes all data aggregation tasks on encrypted data. It then returns and deliver the encrypted aggregated data to the CH through a shortest path covering all intra-cluster-nodes. Because of the large size of a standard WSN, MA-based data collection faces limitations.

A homomorphic encryption based approach for secure data collection is proposed in Hayouni and Hamdi (2017). It aims to provide end-to-end data integrity and confidentiality protection. Homomorphic message authentication codes (HMACs) are used in conjunction with the Elliptic Curve ElGamal algorithm to validate the aggregation results and integrity.

According to Hsieh et al. (2018), homomorphic encryption can be applied in combination with CS, allowing the BS to recover the initial sensory data utilizing the CS restoration algorithm and decode the received messages. They are used in crowdsensing recovery to restore all user trajectories based on trajectory associations, however they are not useful for data gathering in cluster based WSNs.

Zhang et al. (2018b) introduces a secure data gathering scheme built on the combination of CS and homomorphic encryption. This scheme addresses known-plaintext attacks and chosen-plaintext attacks by encrypting the CS output using homomorphic encryption; however, it ignores the CS features.

The study discussed in Ifzarne et al. (2020, 2021) describes a technique for detection and classification of attacks in CS using homomorphic encryption and a machine learning algorithm. The primary goal of this technique is to ensure data security and confidentiality when it is transmitted from nodes to the BS.

To address the security challenge faced by the global encryption key demanded by homomorphic operation, Onion Homomorphic Encryption-based Aggregation (OHEA) was proposed in Tang and Hu (2020). It splits the sources into classes, with each class sharing a specific encryption key (referred as a group encryption key) and transmitting their encrypted results to the leaf-level aggregators. All leaf-level and intermediate-level aggregators have a private key that is used to encrypt the aggregated output before sending it to its parent node. This mechanism propagates to higher levels over and over until it hits the sink.

The differences between the above mentioned schemes and our proposed scheme are summarized in Table 1. The proposed security technique in this paper is based on Homomorphic Encryption as public key algorithm for encrypting the data from the nodes along with trust and energy aware clustering method. In addition, the proposed technique wisely employs the advantage of the CS approach to achieve the data traffic load balancing during the network operation phase to enhance the way the public key algorithms are used in IoT network, to achieve high protection with energy saving.

Table 1
Summary of secure data gathering methods used in IoT-WSN.

Approach	Methodology	Trust	Encryption	Correctness	Data reduction	Performance metrics	Routing methodology
Xiaohan et al. (2019)	Data collection based on ElGamal cryptography and sparse matrices based CS	×	ElGamal cryptography	×	Plain CS	Energy consumption	Grid based
Liu et al. (2019)	Data collection based on public-key cryptography and compressive sensing over finite fields	×	INDCCA1, IND-CPA	×	Plain CS	Ciphertext expansion	Predefined tree based
Zhang et al. (2018a)	Data collection based on CS and asymmetric semi-homomorphic encryption.	×	Asymmetric semi-homomorphic encryption.	✓	Plain CS	Energy consumption, Communication cost, Recovery accuracy	Predefined tree based
Samydurai et al. (2020)	Data Gathering based on Fuzzy scheme.	×	Asymmetric key cryptography	×	×	Propagation delay, Data Gathering Ratio, Control overhead	Predefined cluster based
Kumar et al. (2015)	Data Aggregation based on homomorphic encryption and mobile agent.	×	Homomorphic encryption	×	×	Bandwidth, Energy consumption and delay	Predefined tree based
Hayouni and Hamdi (2017)	Data Aggregation based on homomorphic message authentication codes (HMACs) combined with the Elliptic Curve ElGamal algorithm	×	×	✓	×	Communication overhead and energy consumption	Predefined cluster based
Hsieh et al. (2018)	A cloud-assisted compressive sensing-based data gathering system	×	CS crypto	✓	Plain CS	Reconstruction quality, computation cost	Not specified
Zhang et al. (2018b)	Data Aggregation based on homomorphic encryption	×	Homomorphic encryption	✓	×	Communication cost	Predefined tree based
Ifzarne et al. (2020, 2021)	Data collection based on homomorphic encryption and CS	×	Homomorphic encryption	×	Plain CS	Overhead, Throughput, Delay	Predefined cluster based
Tang and Hu (2020)	Onion Homomorphic Encryption based Aggregation	×	Homomorphic Encryption	✓	×	Delay, Computational and Communication Cost.	Predefined tree based
Salim et al. (2020)	Data gathering based on public key and CS	×	RSA and CS crypto	×	Plain CS	Energy, network lifetime, reconstruction quality	Hybrid
SEEDGT	Data gathering based on integration between trust, public key algorithm, and Compressive Sensing (CS)	✓	Homomorphic Encryption	✓	Modified CS	Energy, network lifetime	New clustering algorithm based trust (called TEECA)

4. SEEDGT: Secure and Energy Efficient Data Gathering Technique for IoT-WSNs

In this section, we illustrate the main principle of the proposed SEEDGT technique, which is divided into three stages: Cluster formation phase, network operation phase and reconfiguration phase. In cluster formation phase, the network is divided in clusters by employing energy aware and trust based method for CH selection. The network operation phase evolves through three stages: Key Sharing and Data Encryption, Encrypted Data Aggregation, and Data Recovery. In the last phase, the network is reconfigured to consider the changes that could occur during network operation. Fig. 1 shows the flow diagram of our SEEDGT technique.

4.1. System model

Here we provide the system model: energy and network models adopted in the proposed work are given as follows:

Network Model: n sensor nodes are distributed at random in an area R with dimension $A \times A$ and a fixed BS. The network was built with the following assumptions in mind:

1. The sensor nodes are capable of self-localization (Silmi et al., 2021; Sabale and Mini, 2021; Lalama et al., 2021).
2. Every node has a unique ID and static.
3. For the purposes of trust assessment, it is assumed that all nodes can listen and interact with their neighbors who lie within their transmission range.
4. Each cluster is handled by a single CH, who is awake throughout a round of operation.
5. Sensor nodes gather environmental data (for example, humidity or temperature) and send it to their CH nodes.

Energy model: We consider the energy expended for data propagation by each node in the cluster as well as the energy utilized for data reception, processing, and transmission from each CH. The energy expended for environmental sensing is not taken into account because they are typically much lower than processing and transmission costs. Moreover, the energy used for data communication is assumed to vary based on the distance between the communicating devices. The energy model used is same as that of Abu Salem and Shudifat (2019), Salim et al. (2020), Vijayalakshmi and Senthilkumar (2020), aghavaraju (2017), Priyadharshini et al. (2021), Rodríguez et al. (2020), Aziz et al. (2021) and Osamy et al. (2019b) because it is the generally recognized model used in several research work, e.g., Abu Salem and Shudifat (2019) and Salim et al. (2020). To transmit a message of $l - bit$ size over a distance d, the radio power expended is:

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2 & d < d_0 \\ lE_{elec} + l\epsilon_{mp}d^4 & d \geq d_0 \end{cases} \quad (2)$$

To obtain this message, the radio expended is:

$$E_{Rx}(l) = lE_{elec} \quad (3)$$

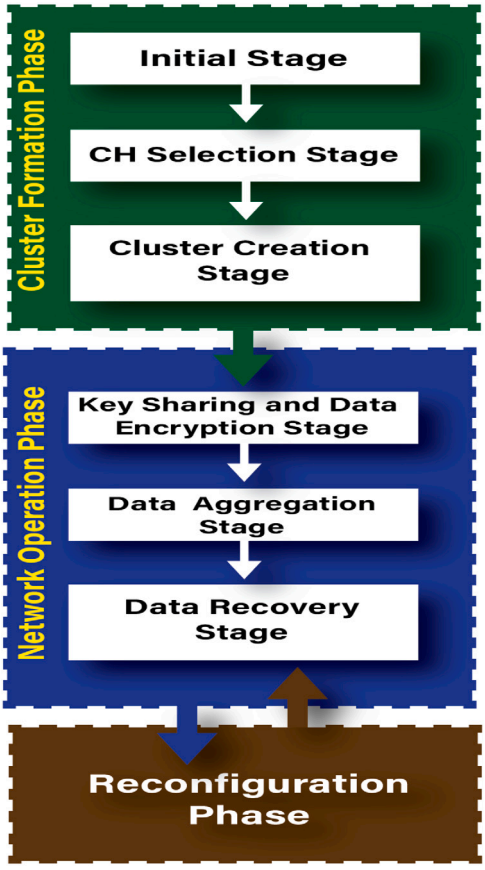

Fig. 1. Flow chart: Overall process flow diagram of SEEDGT.

The parameters used for the simulated model are given as $E_{elec} = 50\ \text{nJ/bit}, \epsilon_{fs} = 10\ \text{pJ/bit/m}^2, \epsilon_{mp} = \frac{13}{1000}\text{pJ/bit/m}^4, d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}$.

In the following subsections, we discuss the selection factors that we employ for cluster creation and trust management processes, and then we explain each phase in detail.

4.2. Selection factors

To mitigate the complexity of network communications, the clustering principle is applied in WSN. The CH, which is chosen as the representative of a cluster is responsible to relay data from the cluster nodes to the BS. However, selecting CH is a tedious process and the selection should be wisely performed for enhancing network lifetime performance and security. Therefore, there is a need to develop a new approach that can effectively choose CH and, at the same time, ensures efficient and secure data collection. To achieve this, we take into account several important factors. The factors that we use for CH selection are defined as follows.

- Residual energy (E): Residual energy is the vital aspect of a sensor node, and network lifetime is primarily determined based on the residual energy of the nodes.
- Distance: This factor is crucial to consider since longer distances need more energy to transmit a data packet. As a result, the total distance between a node and its neighbors should be considered. It is computed using the following equation:

$$ND(s) = \sum_{i \in N_s} d_{s,i} \tag{4}$$

where, $d_{s,i}$ is the node s to neighbor i ($i \in N_s$) distance and ND gives the total distance. Nodes with less ND is preferred to work as CH nodes.

- Trust Factor (TF): Node i evaluates the trust value for one hop neighbors at time t. The proposed technique is based on one-hop, and hence, the node i uses its direct observation with respect to neighbor node j during a round of operation to evaluate trust. The node i applies the following detection factors to make a direct observation and assess the one-hop neighbor node j at a time t. According to our scheme, trust is evaluated in terms of the following three aspects: data forwarding rate, energy rate and success packets rate .
  - Success packets rate (SPR): Because of the wireless nature, there is a possibility that same packets can be received from different sources, i.e. one time directly from the sender and another time from another node for a subsequent transmission. It is found that every packet sent to each node has a timestamp. Using this, a packet can be effectively identified regardless of whether the packets contain similar content. At time t, $SPR(i,j)^t$ of node j with respect to node i (Feng et al., 2011; Rehman et al., 2017) can be given as follows:

$$SPR(i,j)^t = \frac{P_{required}(i,j)^t}{P_{required}(i,j)^t + P_{redundant}(i,j)^t} \tag{5}$$

    $P_{required}(i,j)^t$ and $P_{redundant}(i,j)^t$ denote the required sending packets and the redundant number of packets sent, respectively. The influence of SPR factor is on data forward attack by monitoring the data packets of neighbor nodes.
  - Forwarding rate (FR): Node j can forward data of node i and broadcast ACK. Node i can collect node j's ACK to determine the sending packets count. $FR(i,j)^t$ of node j with respect to node i at time t (Feng et al., 2011; Rehman et al., 2017) can be given as follows:

$$FR(i,j)^t = \frac{FR(i,j)^t - FR(i,j)^{t-1}}{FR(i,j)^t + FR(i,j)^{t-1}} \tag{6}$$

    The influence of FR factor is to monitor malicious node activity and provide protection against sinkhole attacks.
  - Energy Trust Rate or Energy Confidence (ETR): For sensor nodes, energy is a critical factor. This represents the sensor nodes' ability to act normally. Since the nodes' power is limited, some malicious nodes refuse to cooperate in order to conserve power (Saidi et al., 2020). The ETR is given by

$$ETR(i,j) = \frac{E(j)^t}{E^t_{avg}} \tag{7}$$

    where $E(j)^t$ is the consumed energy of node j at time t and E^t_{avg} is the average power consumption at time t.
    E^t_{avg} is used to prevent malicious node from sending incorrect or fake information about its remaining energy. We may conclude whether or not the node has abnormal energy usage based on the ratio change. The value '1' indicates that the node has a regular rate, and less than '0.5' or greater than '1' indicates that the node has an abnormal rate, which triggers it to be blocked. This aspect is critical during round operations, as it prevents malicious nodes from sharing the CH selection mechanism in subsequent rounds.

The trust factor of node i that categorizes the behavior of node j is a combination of SPR, FR, and ETR. The trust factor (TF) can be estimated as follows:

$$TF(i,j)^t = w_1 \times SPR(i,j)^t + w_2 \times (1 - FR(i,j)^t) + w_3 \times ETR(i,j)^t \tag{8}$$

The parameters w_i, $1 \le i \le 3$ are the weights, $w_1 + w_2 + w_3 = 1$. We assign the same weight to all the trust factors.

4.3. Cluster formation phase

In this section, we introduce our proposed Trust and Energy Efficient Clustering Algorithm (TEECA) for CH selection and cluster formation. Following deployment, BS broadcasts a message to all network nodes. When this message is received, each node computes its distance to BS. Then, in order to identify its neighbors, each node broadcasts a $Hello$ message. Each node gathers information about its neighbors during this first stage of communication. This information is useful for monitoring node behavior, since nodes can measure the direct trust value of monitored nodes via direct information communication, and nodes can indirectly calculate the indirect trust value of monitored nodes via common neighbors through indirect information communication. The trust factor is significant because it indicates whether a node is trustworthy or not. In order to form network clusters, the following stages are performed once.

4.3.1. Initial stage

After node deployment, each node has information about its neighbors. Then each node calculates the trust with respect to its neighbor node using Eq. (8). Each node transmits the estimated values to the BS, which aggregates them and determines the final trust values before determining whether the nodes are trustworthy or not, based on user-defined thresholds. The BS send its feedback trust value ($TF_{feedback}, 0 < TF_{feedback} \leq 10$ and it will be used later in cluster election process) along with blacklist nodes (list of nodes that have malicious behavior) to each node.

4.3.2. CH selection stage

The next step is for each node s to choose a random number $s.t_r, 0 \leq s.t_r \leq 1$, which is then compared to the threshold value T. Nodes whose generated $s.t_r$ is greater than T serve as CHs, while nodes whose generated $s.t_r$ is less than T become members. Therefore, the calculation of threshold T in Abu Salem and Shudifat (2019) is refined by taking into account the weight value $s.U$ for each node s, based on distance, trust feedback and energy values.

The weight value is determined as follows:

$$s.U = \alpha \times F(E_{remaining}) + \beta \times (1 - F(s.ND)) + \gamma \times F(TF_{feedback}(s)) \quad (9)$$

where α, β and γ are the weight coefficients, with sum of $\alpha + \beta + \gamma = 1$, and are modified based on the application's specifications and security level. The estimations of coefficients should be decided on the basis of the relevance of each measure in the WSNs applications under consideration (Abu Salem and Shudifat, 2019; Castillejo et al., 2015).

$TF_{feedback}(s)$ is the feedback trust value for node s, $F(x) = 1 - \frac{1}{\delta + x}$, and $\delta > 0$ is a tunable constant. The feature F has the attractive property of rapidly approaching 1 as δ increases. Any other function with the property of rapidly approaching 1 can be employed; however, the selection of this function is due to its simplicity, we suggest that $\delta = 1$.

The threshold value for a node s is determined finally as follows:

$$s.T = \begin{cases} \frac{p \times s.U}{(1 - p \times mod(t, \lceil p^{-1} \rceil))} & \text{if } \omega(s,t) = 0 \\ 0, & \text{if } \omega(s,t) > 0 \end{cases} \quad (10)$$

where, p represents the CHs % and $\omega(s,t)$ is a function that specifies whether or not the node s should share the selection process at round t; a zero value indicates that s can serve as CH at round t, while a non-zero value indicates that s cannot act as CH at round t. Initially $\omega(s,t) = 0$ for all nodes and it is updated using Eq. (11), if it becomes CH then it resets to zero if $\mod(t, round(1/p))$ equals to zero.

$$\omega(s,t) = round(\frac{1}{p}) - 1 \quad (11)$$

at the end of this stage, the cluster creation step is executed as follows:-

4.3.3. Cluster creation stage

Following the selection of the CH, cluster formation occurs as follows: Every non-CH node s joins a CH (CH_i) that meets the three criteria: (1) CH_i is not included in the blacklist; (2) the distance from s to CH_i is less than the distance from s to BS; and (3) the distance from CH_i to BS is less than the distance from s to BS. If any of the three requirements listed for the node s do not match with any of the chosen CHs, it chooses the nearest CH that is not on the blacklist.

Algorithm 4 Trust and Energy Efficient Clustering Algorithm (TEECA)

```
CHs: Set of selected cluster heads.
p: Percentage of cluster heads.
at node side:
for each node s_i do
    Compute trust value for each neighbor nodes using Eq. (8)
    Send computed values to BS
end for
At BS side:
Aggregate all trust values
Add nodes with values less threshold to blacklist BL.
Assign feedback value for each node
Send BL and feedback to the network
At node side:
for each node s_i do
    s_i.state ← plain
    Calculate s_i.U using Eq. (9)
    {Selection process}
    if ω(s_i, t) == 0 And s_i ∉ BL then
        s_i.t_r= rand()
        Compute s_i.T using Eq. (10)
        if s_i.t_r < s_i.T then
            s_i.state ← head
            CHs = CHs ⋃ s_i
        end if
    end if
end for
{Cluster creation}
for each CH in CHs do
    Broadcast advertisement message (AD_Message)
    Wait for join messages Join_Request that includes node information
end for
for each node s_i & s_i.state not CH do
    Set head for s_i to BS
    for each CH in CHs and CH ∉ BL and s_i received advertisement message from CH
    do
        if d(CH, BS) < d(s_i, BS) and d(s_i, CH) < d(CH, BS) then
            Send Join_Request message to CH
            Set head for s_i to CH
        end if
    end for
    if AD_Messages received and s_i.Head == BS then
        From the received AD_Messages, select and send join message to the nearest
        CH ∉ BL.
        s_i.Head = CH
    end if
end for
```

4.4. Network operation phase

The proposed technique aims the security of the IoT-WSN network with less energy consumption. A direct way to achieve this aim is to use Homomorphic Encryption system besides using the CS method to compress and reduce the data size of nodes in order to reduce the power expended during the transmission process. Moreover, the proposed technique utilizes the aggregation property of Homomorphic system to aggregate the data on the CH side and (in contrast to other public key algorithms) eliminates the need to decrypt them. This phase divided into three stages: Key Sharing and Data Encryption, Data Aggregation, and Data Recovery. First we describe our proposed — modified Plain CS method, and then describe each stage in detail. The working of network operation phase is provided in Algorithm 5.

In terms of utilizing the CS method in order to compress and reduce the sensors data, in this paper, modified Plain CS method is proposed to utilize the CS in an efficient way and achieve energy saving. In plain CS, each sensor generates the corresponding coefficient of CS matrix for data compression and subsequently sends M samples to the CH. The CH

then uses CS to compress the received samples and transmits M samples to the BS as illustrated in Fig. 2(a). Even if the data size is less than M, each node transmits M samples, which leads to unwanted high traffic at the early stages of transmission. Thus, the proposed modified method aims to address the plain CS problem by preventing the CM which is located in a cluster with size less than M to compress their data but sending the data directly without compression. On the other hand, the CM which is located in cluster with size $\geqslant M$ compresses the data using CS and transmits M to the CH. Using the proposed modified plain CS method, we can successfully reduce the randomness of using CS with all sensors even when data is less than M. To prove the efficiency of proposed modified CS method in terms of energy consumption, we provide the following energy analysis.

4.4.1. Energy analysis for the modified plain CS method

Consider the radio energy communication model presented in Section 4.1. Suppose N sensor nodes are randomly placed in a rectangle area A. The nodes are grouped into K clusters with one CH (CH_i) and a number of CMs $|C_i|$ in each cluster i. Hence,

$$\sum_{i=1}^{K} |C_i| + K = N$$

Each CM captures and delivers data to its CH at the finish of each round. CH takes the responsibility of sending the collected data towards the BS. The total energy consumed by CMs for cluster i is:

$$E^i_{CM} = m_1 E_{Tx_1} + m_2 E_{Tx_2} + \cdots + m_j E_{Tx_j} + \cdots + m_{|C_i|} E_{Tx_{|C_i|}}. \tag{12}$$

Here, m_j is the count of data packets required to send data vector D_j of node j to the respective CH_i. E_{Tx_j} is transmission energy expended by node j. The total energy expended by each CH (CH_i) is given as follows:

$$E^i_{CH} = \sum_{j=1}^{|C_i|} m_j E_{Rx} + (m_i + \sum_{j=1}^{|C_i|} m_j) E_{Tx_i}. \tag{13}$$

Therefore, the overall expended energy for a cluster (i) is:

$$E^i_{Total} = E^i_{CM} + E^i_{CH}. \tag{14}$$

Consider all nodes have the same starting energy E_0. Let d_{CH} is the average distance from CM to CH, d_{BS} is the average distance from CH to BS, $E_{Tx}(b, d_{CH})$ is the energy expended by each CM to send b bits packet to its CH and $E_{Tx}(b, d_{BS})$ is the expended energy to send b packets to BS by CH, respectively. As a result, Eq. (12) can be reduced as follows:

$$E^i_{CM} = \sum_{j=1}^{|C_i|} m_j E_{Tx}(b, d_{CH}). \tag{15}$$

The total expended energy for K clusters in each round, without any compression or aggregation mechanism, can be given as follows:

$$E_{Total} = \sum_{i=1}^{K} (\sum_{j=1}^{|C_i|} m_j E_{Tx}(b, d_{CH}) + \sum_{j=1}^{|C_i|} m_j E_{Rx} + (m_i + \sum_{j=1}^{|C_i|} m_j) E_{Tx}(b, d_{BS})). \tag{16}$$

Generally, in the plain CS technique, CMs use the CS method to send data to the CHs, who subsequently receive the measurement and send the aggregated results to the BS. Referring to the CS technique and Eq. (16), the overall energy spent by K clusters every round is:

$$E^{CS}_{Total} = \sum_{i=1}^{K} (M\ |C_i| E_{Tx}(b, d_{CH}) + M\ |C_i| E_{Rx} + ((M + M\ |C_i|) E_{Tx}(b, d_{BS}))). \tag{17}$$

When compared to Eq. (16), data traffic communication can be reduced when:

$$(\sum_{i=1}^{K} |C_i| + 1) M \ll \sum_{i=1}^{K} (m_i + \sum_{j=1}^{(|C_i|)} m_j) \tag{18}$$

and

$$M \sum_{i=1}^{K} |C_i| \ll \sum_{i=1}^{K} \sum_{j=1}^{|C_i|} m_j \tag{19}$$

From the previous Eqs. (17), (18), and (19), we can notice that the cluster size has great influence on applying CS in an efficient way. We have two cases: worst case and best case. The worst case is when cluster size is less than M and each node sends M samples even if its data is less than M, resulting in unneeded extra traffic (see Fig. 2(a)). While, the best case is when cluster size is larger than or equal to M (see Fig. 2(b)). In the proposed modified CS method, we employ the CS method considering the cluster size which leads to save energy by removing unnecessary traffic caused by the plain CS method.

4.4.2. Key sharing and data encryption stage

In our proposed technique, the BS uses Algorithm 1 to produce the public (k_p) and the private (k_{pr}) keys for data encryption and decryption, respectively. Then, the BS broadcasts the global seed for CS matrix generation only to clusters where there are $\geq m$ number of CM, followed by the public key k_p to the entire network. This instance will only be used once before any transmission from either side. The proposed scheme assumes the following based on the clusters size in order to adopt the proposed modified plain CS with Homomorphic method:

1. If the Cluster size $|Cl_j| = A$ is smaller than the CS measurement (M), i.e., $A < M$, then each CM will encrypt its data using Homomorphic Public Key Algorithm 2 and send the cipher text data to CH as follows: each node a such that $1 \leq a \leq A$ uses the public key $k_{pr} = (g, n)$ to encrypt its data $E(d_a)$ into c_a such that $c_a = g^{d_a} \times r_a mod\ n$, where r_a is a random number, and then sends the encrypted data c_a to the CH. After that, the CH performs aggregation of the cipher text data using Homomorphic aggregation method in Eq. (1) and then the aggregated cipher text is sent to the BS.
2. If the cluster size $|Cl_i| = B$ is larger than or equal to M, *i.e.*, $B \geq M$, then each CM b such that $1 \leq b \leq B$ uses the received global seed from the BS along with its cluster member's id to construct the corresponding CS sub-matrix Φ_b in order to compress its data such that $y_b = \Phi_b d_b$. After that, each CM b uses Algorithm 2 to encrypt the compressed data $E(y_b)$ such that $E(y_b) = c_b = g^{d_b} \times r_b\ mod\ n$, where r_b is a random number, and then sends the encrypted data c_b to the CH. After that, the CH performs aggregation of the cipher text data using Homomorphic aggregation method in Eq. (1) and then the aggregated cipher text is sent to the BS.

4.4.3. Data aggregation stage

At this stage, the proposed technique strives to overcome the drawback of public key algorithms that do not permit the CH to perform data aggregation process. This problem is addressed by the proposed technique using Homomorphic aggregation method in Eq. (1) as follows:

Case 1: For the cluster Cl_j ***with size*** $|Cl_j| = A$, $A < M$, ***each CH will do the following:***

1. Collect ciphered vector $[c_1, c_2, \ldots, c_A]$ where $[c_1, c_2, \ldots, c_A] = [g^{d_1} \times r_1 \mod n, g^{d_2} \times r_2 \mod n, \ldots, g^{d_A} \times r_A \mod n]$ from member nodes such that $c_1 = E(d_1), c_2 = E(d_2) \ldots, c_A = E(d_A)$, where d_i is the original data for node i.

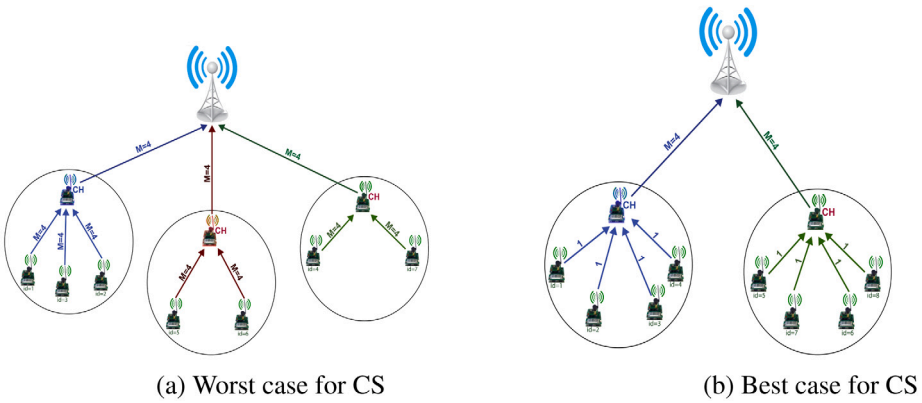

Fig. 2. CS method for data aggregation in IoT-WSN.

2. Use Eq. (1) in order to aggregate them into C_j such that:

$$\begin{aligned} C_j = c_1 \times c_2 \times c_3 \times \cdots \times c_A \mod n \\ = g^{d_1} \times r_1 \mod n, g^{d_2} \\ \times r_2 \ mod \ n, \ldots, g^{d_A} \times r_A \ mod \ n \\ = g^{d_1+d_2+\cdots+d_A} \times r \ mod \ n \\ = g^D \times r \ mod \ n, where \ D_A \\ = [d_1 + d_2 + \cdots + d_A] \end{aligned}$$

3. Send C_j to the BS

Case 2: For the cluster with size $|Cl_i| = B$ ***bigger than or equal to*** M, ***i.e.,*** $B \geq M$, ***each CH does:***

1. Collect ciphered vector $[c_1, c_2, \ldots, c_B]$ where $[c_1, c_2, \ldots, c_B] = [g^{y_1} \times r_1 \mod n, g^{y_2} \times r_2 \mod n, \ldots, g^{y_B} \times r_B \mod n]$ from member nodes, such that $[c_1 = E(y_1), c_2 = E(y_2), \ldots, c_b = E(y_b), \ldots, c_B = E(y_B)]$, where y_b is the compressed data for node b.
2. Use Eq. (1) to aggregate them into C_i such that:

$$\begin{aligned} C_i = c_1 \times c_2 \times c_3 \times \cdots \times c_B \mod n \\ = g^{y_1} \times r_1 \mod n, g^{y_2} \\ \times r_2 \mod n, \ldots, g^{y_B} \times r_B \mod n \\ = g^{y_1+y_2+\cdots+y_B} \times r \ mod \ n \\ = g^Y \times r \mod n, where \\ Y = [y_1 + y_2 + \cdots + y_B] \end{aligned}$$

3. Send C_i to the BS

According to this scenario, the CH does not need a private key to decrypt the cipher data to aggregate the data of the cluster members. This leads to a reduction in communication costs and reduces overall energy consumption compared to the direct application of the public key technology.

Consider the following example to clarify the above: Assume that we have a network that consists of $N = \{n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8, n_9, n_{10}\}$ nodes. It is divided into two clusters C_1 where node n_1 is CH and nodes $\{n_2, n_3, n_4, n_5, n_6, n_7\}$ are the CM, i.e., size of $|\ C_1\ | = 7$, and C_2 where node n_8 is CH and nodes $\{n_9, n_{10}\}$ are the CM, i.e., size of $|\ C_2\ | = 3$. CS measurement size $|\ m\ | = 4$. The BS sends k_p and the seed of the CS matrix to the network. Since $|\ C_2\ | < m$, each CM node $\{n_9, n_{10}\}$ uses k_p and Algorithm 2 to encrypt its data and calculate the corresponding cipher data $\{c_9, c_{10}\}$, and finally sends $\{c_9, c_{10}\}$ to the CH, which calculates c_8 then uses Eq. (1) to aggregate all of $\{c_8, c_9, c_{10}\}$ and sends the aggregated cipher data to the BS.

On the other hand, since $|\ C_1\ | > m$, each CM nodes, i.e., $\{n_2, n_3, n_4, n_5, n_6, n_7\}$ uses the CS matrix seed to construct the CS matrix that is used to compress its data and calculate the corresponding CS measurement vectors $\{y_2, y_3, y_4, y_5, y_6, y_7\}$. Then each node uses k_p and Algorithm 2 to encrypt its data and calculate the corresponding cipher data i.e $\{c_2, c_3, c_4, c_5, c_6, c_7\}$ and finally sends $\{c_2, c_3, c_4, c_5, c_6, c_7\}$ to the CH, which calculates y_1 using the CS matrix and c_1 using k_p. After that, CH uses Eq. (1) to aggregate all of $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$ and sends the aggregated cipher data to the BS.

4.4.4. Data recovery stage

At this stage, the BS receives two types of data: ciphered compressed samples $C_i = g^Y \times r \ mod \ n$ and ciphered uncompressed data $C_j = g^D \times r \mod n$ such that $i \neq j$. The BS differentiates between them on the basis of its source: if it has come from CH where its cluster size $\geq m$, then it is the ciphered compressed sample, otherwise it is ciphered uncompressed data. The BS will do the following in order to obtain the original data:

For the C_i, ***the BS does:***

1. Use Algorithm 3 and the private key k_{pr} to decrypt the cipher data into $Y = [y_1 + y_2 + y_3 + \cdots + y_B]$, based on Eq. (1):

$$\begin{aligned} D(C_i) = D(g^Y \times r \mod n) \\ = D(E(y_1) \times E(y_2) \times \cdots E(y_B)) \\ = y_1 + y_2 + y_3 + \cdots + y_B \end{aligned}$$

Where $Y = y_1 + y_2 + y_3 + \cdots + y_B$ is the CS aggregation vector.

2. Apply the same global seed to produce the CS matrix Phi, to solve the $\|l\|_1$-norm minimization problem as given below:

$$D_B = argmin\|D_B\|_1, subject \ \ to \ \ y = \Phi D_B \tag{20}$$

where D_B is vector which contains CM original data.

3. Within the CS framework, Eq. (20) can be solved efficiently using any greedy strategy such as Orthogonal Matching Pursuit (OMP) in Tropp and Gilber (2007), Subspace Pursuit in Wei and Olgica (2009), or COSAMP in Needell and Tropp (2009) to recreate the data from only m.

For the C_j, ***the BS does:***

1. Use Algorithm 3 and the private key k_{pr} to decrypt the cipher data into $D_j = [d_1 + d_2 + d_3 + \cdots + d_A]$, based on Eq. (1):

$$\begin{aligned} D(C_i) = g^D \times r \mod n \\ = D(E(d_1) \times E(d_2) \times \cdots E(d_A)) \\ = d_1 + d_2 + d_3 + \cdots + d_A \end{aligned}$$

Where $D_j = d_1 + d_2 + d_3 + \cdots + d_A$ is the original data aggregation vector.

Algorithm 5 outlines the procedures of Network operation phase of the proposed SEEDGT technique.

Algorithm 5 Network Operation Phase

Key Sharing and Data Encryption Stage:
BS create the public key k_p, the private key k_{pr} and the CS matrix global seed.
BS sends the global seed to the clusters where its size $\geq m$ and broadcast k_p to the entire network.
for each Cluster i do the following **do**
if $|\, Cluster_i \,| \geq m$ **then**
for each node j do the following **do**
Use the global seed and cluster member's id to generate the CS sub-matrix Φ_j.
Use Φ_j to compress d_j vector into m samples such that $Y_j = \Phi_j X_j$.
use k_p to encrypt Y_j into cipher data c_j using Algorithm 2.
send c_j to the CH.
end for
end if
if $|\, Cluster_i \,| < m$ **then**
for each node j do the following **do**
Use k_p to encrypt d_j into cipher data c_j using Algorithm 2.
Send c_j to the CH.
end for
end if
end for
Encrypted Data Aggregation Stage:
if $|\, Cluster_i \,| \geq m$ **then**
for each CH of the cluster i do the following **do**
collect the cipher data from its cluster members into vectors c_i using (1).
end for
end if
if $|\, Cluster_i \,| < m$ **then**
for each CH of the cluster i do the following **do**
collect the cipher data from its cluster members into vectors c_j using (1).
end for
end if
Data recovery stage:
BS applies the same global seed to produce the CS matrix Φ.
if $|\, Cluster_i \,| \geq m$ **then**
BS uses the private key k_{pr} and Algorithm 3 to decrypt vector C_i, then obtains the compressed sample vector Y such that $D(C_i) = y_1 + y_2 + y_3 + \ldots + y_i = Y$
BS uses any reconstruction algorithm to solve Eq. (20) in order to obtain D_i from Y, where D_i the original sensors of cluster i member.
end if
if $|\, Cluster_i \,| < m$ **then**
BS uses the private key k_{pr} and Algorithm 3 to decrypt vector C_j, then obtains the original data vector D such that $D(C_i) = d_1 + d_2 + d_3 + \ldots + d_j = D_j$
BS uses any reconstruction algorithm to solve Eq. (20) in order to obtain D_j, where D_j the original sensors of cluster j member.
end if

4.5. Reconfiguration phase

The behavior of nodes alters over time throughout the subsequent rounds of election. Furthermore, while computing node weight, a high weight is applied to the trust value. As a result, nodes with a high trust value have a better probability of being chosen as CH than nodes with a low trust value.

This phase is executed at the end of each round and following the network operation phase. Each CH_i calculates trustworthiness of member nodes using Eq. (8) and asks each cluster member node to calculate the trustworthiness value about each neighboring node within the cluster and transmits to the CH. The CH then forms the trustworthiness matrix TM for its cluster. The TM matrix is given by

$$\mathrm{TM} = \begin{bmatrix} - & TF(1,2) & TF(1,3) & \ldots & TF(1,m) \\ TF(2,1) & - & TF(2,3) & \ldots & TF(2,m) \\ & & \ldots\ldots\ldots\ldots\ldots\ldots & & \\ TF(m,1) & TF(m,2) & TF(m,3) & \ldots & - \end{bmatrix}$$

Then, CH_i compute the total aggregated trust value about each node j in the cluster using the following equation:

$$CH_i.TF(j) = \lceil 10 * (\frac{TF(CH_i, j) + \frac{\sum_k^m TM(j,k)}{m}}{2}) \rceil \quad (21)$$

An unsigned integer between 0 and 10 only needs 4 bits of memory space, thus saving 50% memory space compared with trust values represented as an integer between 0 and 100 (1 bytes) and 87.5% compared with trust values represented as a real number (4 bytes) (Li et al., 2013).

The weight value of each member node j for CH_i is calculated as follows:

$$U(j) = \alpha \times F(j.E_{remaining}) + \beta \times (1 - F(j.ND)) + \gamma \times F(CH_i.TF(j)) \quad (22)$$

where, $F(x) = 1 - \frac{1}{1+x}$.

The member node j with higher weight value $U(j)$ will be selected as next CH. When more than one node has the same weight value, the CH selection depends on each member node information (refer to Eq. (10)) to randomly select the eligible node in the cluster after resetting its percent of CH to $p = 1/m$ (m is the number of cluster members).

4.6. Computational complexity

According to the proposed technique, the network has two types of clusters: clusters with a size of $< M$ and clusters with a size of $\geqslant M$. Based on this, the computational complexity of the proposed technique can be calculated as follows:

Case 1: cluster size $< M$:

- Each member node s requires one encryption operation. According to the Paillier cryptosystem, each encryption process involves one exponentiation, one modulus and one multiplication operation. The computational complexity of one encryption process is $O(logn)$ (when $n \ll m$, $n = pq$, p and q are two large prime numbers and m is a message) for multiplication operations. Hence, the computational complexity for a CM (s) is $O(logn)$ with respect to multiplication tasks.
- For the CH node, L multiplication aggregation actions should be performed between the received data vector and its own encrypted data vector. As a result, the computational complexity of the CH node is $O(logn + L)$, where L is the cluster size.

Case 2: cluster size $\geq M$:

- Each member node s requires one encryption operation. According to the Paillier cryptosystem, each encryption process involves one exponentiation, one modulus and one multiplication operation. The computational complexity of one encryption operation is $O(logn)$ (when $n \ll m$, $n = pq$, two large prime numbers p and q and m is a message) for multiplication operations. Therefore, The computational complexity on member node is $O(logn)$ with respect to multiplication tasks.
- For CH, M multiplication aggregation actions should be performed between the received data vector and its own encrypted data vector. As a result, the computational complexity of the CH node is $O(logn + M)$.

Since $M > L$, the overall computational complexity on CH node is $O(logn + M)$

To retrieve the original data at the BS, the BS will first decode the encrypted M messages, and then use OMP to rebuild the original data. M decryption operations and one restoration operation utilizing OMP are required by the BS. Decrypting an element in the Paillier cryptosystem involves one exponentiation, one multiplication, and one division operation. In terms of multiplication operations, the computational cost of one decryption operation is $O(logn)$. As a result, decryption has a computational complexity of $O(Mlogn)$. The computational complexity of utilizing OMP to rebuild the actual message $x(t)$ is given by $O(KlogN)$, where K is the sparsity of $x(t)$. The computational cost of retrieving the actual data at the BS is thus $O(Mlogn) + O(KlogN)$.

It is clear that the computational complexity at the BS side is much higher than that of the CH side. That is why the proposed technique shifted the complex operations such as reconstruction process to the BS side which has no power constraints.

4.7. Communication complexity

The proposed SEEDGT scheme has three phases, Cluster formation phase, network operation phase and reconfiguration phase. The complexity of messages exchanged will be calculated for each stage, and the overall complexity of SEEDGT will be calculated as follows:

- **Cluster formation phase complexity:** This includes three stages: Initial Stage, CH selection stage, and Cluster creation stage. Following the deployment of nodes, the Initial Stage is only performed once. It is started by BS and will be progressively created by receiving messages about trust values and other information from every node in the network; hence, the cost of this phase will be n, where n is the number of nodes in the WSN. In CH selection stage, k CHs are selected and each CH broadcasts an advertisement message, while in Cluster creation stage, the k clusters will be constructed by exchanging $n-k$ join messages of CMs. Finally, k CHs send their schedules to the CMs. As a result, the total number of messages exchanged during the setup phase will be $n + (n-k) + 2k = O(n)$, where $k < n$.
- **Network operation phase complexity:** This includes three stages: Key Sharing and Data Encryption, Encrypted Data Aggregation, and Data Recovery.
  - In Key Sharing and Data Encryption stage, each node transmits its encrypted data to its CH. Hence, $n-k$ messages from CMs to their CHs in total.
  - In Data Aggregation stage, each CH delivers the encrypted aggregated data to the BS, resulting in k messages exchanged at this step.
  - In Data Recovery stage, no message exchanges occur.
- **Reconfiguration phases complexity:** In this phase, $n-k$ messages from CMs to their CHs and k messages from each CH to the new elected CH. Therefore, the total message exchanges will be $k + (n-k) = O(n)$.

Therefore, the overall message exchanges for the proposed scheme will be $O(n)$.

4.8. Security analysis

In this section, the proposed security scheme is analyzed in terms of the following security parameters:

- **Correctness:** refers to whether the proposed scheme can acquire the aggregated vector when the homomorphic encryption scheme is used.
- **Data confidentiality:** refers to the prevention of the disclosure of information to unauthorized people, resources, or processes.
- **Trust:** refers to the accuracy, consistency, and trustworthiness of data resources.

The following lemmas prove that the proposed security scheme successfully achieves the above mentioned security parameters.

Lemma 1. *The use of homomorphic encryption data aggregation technique in the proposed scheme ensures that the obtained aggregated data at the BS after decryption is the aggregated result of the encoded data of all nodes.*

Proof. The proposed scheme has two types of clusters: (1) clusters Cl_j with size less than M such that $|Cl_j| = A < M$, and (2) clusters Cl_i with size bigger than or equal to M such that $|Cl_i| = B \geq M$. In the first type, each node a such that $1 \leq a \leq A$ encrypts its data using Homomorphic public key and then sends the Cipher $C_a = E(d_a)$ to the CH, where d_a is the original data. Then, each CH collects $C_j = c_1 \times c_2 \times c_3 \times \cdots \times c_A \mod n$ using homomorphic aggregation method and sends it to the BS. In the second type, each node b such that $1 \leq b \leq B$ uses the global seed to compress its data and generate y_b, encrypts the y_b using the public key to obtain the cipher data $c_j = E(y_j)$ and then sends the cipher data to the CH. Then, the CH collects $C_i = c_1 \times c_2 \times c_3 \times \cdots \times c_B \ mod n$ using homomorphic aggregation method and sends it to the BS. Finally, the BS uses the private key to decrypt the C_j and C_i and obtain $D(C_j) = D(E(d_1) \times E(d_2) \times \cdots \times E(d_A)) = d_1 + d_2 + d_3 + \cdots + d_A = D_j$ and $D(C_i) = D(E(y_1) \times E(y_2) \times \cdots E(y_B)) = y_1 + y_2 + y_3 + \cdots + y_B$, then uses the CS matrix and CS reconstruction algorithm to generate D_i where D_i is the vector that contains CM's original data. Thus, according to Lemma 1, the BS finally obtains the whole network data $D = \sum_{i=1}^{N} D_i$ after it generates D_i and D_j.

Lemma 2. *The proposed scheme achieves data confidentiality.*

Proof. The proposed scheme uses the power of the implemented public key by homomorphic public key algorithm to achieve the data confidentiality. Homomorphic public key algorithm does not need the BS to share the private key with CHs to operate the aggregation process, and it uses the homomorphic aggregation process to allow the CHs to aggregate data without decrypting the cipher data. I.e., the BS is the only node that can decrypt the network data because it is the only node that has the private key.

Lemma 3. *The proposed scheme allows each node to communicate only with trustable nodes and prevents the untrustable nodes to act as CH.*

Proof. The proposed Trust and Energy Efficient Clustering Algorithm (TEECA) for CH selection and cluster formation monitors node behavior, where nodes can measure the direct trust value of monitored nodes via direct information communication, and nodes can indirectly calculate the indirect trust value of monitored nodes via common neighbors through indirect information communication. The trust factor is significant because it indicates whether a node is trustworthy or not in order to prevent the compromised nodes from accessing the privileged information. Furthermore, each node only authenticates and communicates with the trusted node, preventing internal attacks.

Based on the previously discussed security parameters, the proposed scheme can successfully protect the IoT-WSN network from both internal and external threats utilizing the proposed trust management algorithm and the public key algorithm, respectively. Moreover, the proposed scheme utilizes the aggregation property of homomorphic encryption scheme during the cluster data gathering and integrates it with CS method to achieve strong security performance and load balancing through the use of public key algorithm as well as data vector size reduction through the use of the proposed modified CS method.

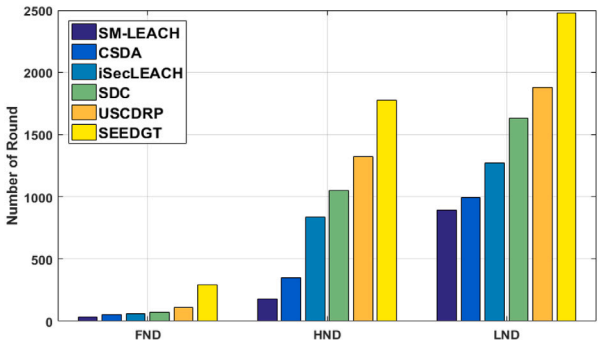

Fig. 3. FND, HND and LND in SEEDGT, SDC, CSDA, iSecLEACH, SM-LEACH and USCDRP algorithms.

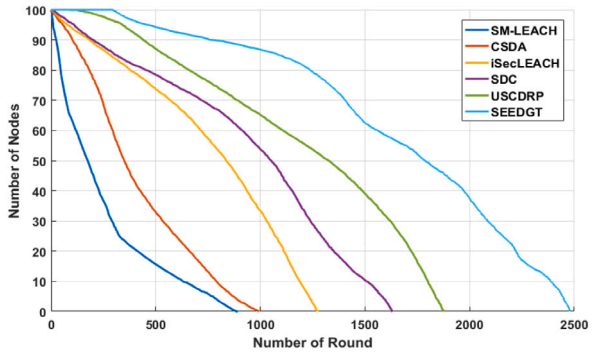

Fig. 4. Alive nodes count per round in SEEDGT, SDC, CSDA, iSecLEACH, SM-LEACH and USCDRP algorithms.

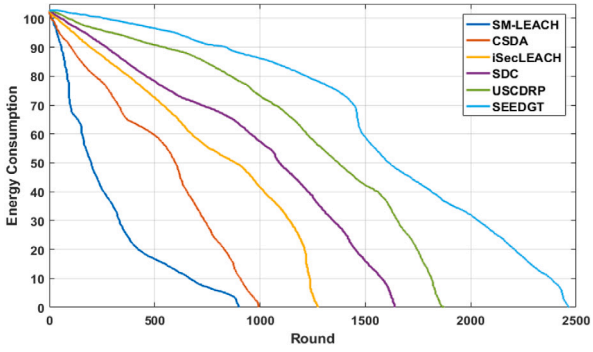

Fig. 5. The energy per round in SEEDGT, SDC, CSDA, iSecLEACH, SM-LEACH and USCDRP algorithms.

5. Simulation results and discussion

Here, we assess the performance and findings of the proposed SEEDGT scheme simulated in MATLAB R2015a. First, SEEDGT's performance is analyzed and compared with SDC (Ifzarne et al., 2021), CSDA (Fang et al., 2019), iSecLEACH (aghavaraju, 2017), SM-LEACH (Priyadharshini et al., 2021), and USCDRP (Vijayalakshmi and Senthilkumar, 2020) algorithms, where, iSecLEACH, SMLEACH, and USCDRP are public-key based secure data gathering schemes, and SDC, CSDA are secure data gathering schemes based on CS. Then, the proposed TEECA algorithm is assessed and compared with baseline algorithms (Trust based clustering schemes) such as: EESTBCA (Rehman et al., 2017), HiTSeC (Gaber et al., 2018), Trusted-LEACH (Narayan et al., 2019) and LEACH-TM (Fang et al., 2021). The selection of these schemes for comparison is for the reason that they are related to our main goal. Finally, the effect of transmission radius on our SEEDGT is discussed.

5.1. Evaluation of SEEDGT scheme

In this section, we employed the same simulation conditions as in Vijayalakshmi and Senthilkumar (2020), where the network dimension is 100 m × 100 m with 100 nodes, and the BS is located in the network's center. We assign the same weights to the trust factors w_1, w_2, and w_3 in Eq. (8) and same weights to the weight factors α, β, and γ in Eq. (9). The performance of SEEDGT is evaluated and compared with SDC (Ifzarne et al., 2021), CSDA (Fang et al., 2019), iSecLEACH (aghavaraju, 2017), SM-LEACH (Priyadharshini et al., 2021) and USCDRP (Vijayalakshmi and Senthilkumar, 2020) algorithms. The performance metrics utilized for evaluation are as follows:

1. Network lifetime (First (FND), Half (HND) and Last node (LND) dead).
2. Count of alive nodes per round.
3. Average network energy per round.

Fig. 3 shows the lifetime of the network (FND, HND and LND respectively). The count of alive nodes per round for SEEDGT, SDC, CSDA, iSecLEACH, SM-LEACH and USCDRP algorithms are depicted in Fig. 4. It is clear that SEEDGT has a higher number of alive nodes every round than SDC, CSDA, iSecLEACH, SM-LEACH and USCDRP algorithms. From Figs. 3 and 4, it is evident that SEEDGT improves the lifetime of WSN in terms of the FND compared with SDC, CSDA, iSecLEACH, SM-LEACH and USCDRP algorithms. This is because SEEDGT used the Homomorphic aggregation method which allows the CH to aggregate the encrypted data without the need to decrypt them, resulting in significant computational power savings. In addition, the proposed algorithm utilizes the CS method to attain energy load balancing among the nodes.

In Fig. 5, we analyze the energy consumption of SEEDGT, SDC, CSDA, iSecLEACH, SM-LEACH and USCDRP algorithms. Fig. 5, shows that SEEDGT has minimum energy consumption per round than the others, meaning that in each round, SEEDGT consumes comparatively less energy than SDC, CSDA, iSecLEACH, SM-LEACH and USCDRP algorithms. This is because SEEDGT uses the modified CS method which efficiently utilizes the CS method in order to remove the unnecessary higher traffic at the beginning stage of transmissions.

Based on the above results, we can conclude that SEEDGT outperforms all other techniques in terms of achieving security with the best network lifetime performance. The main reason for these results is that SEEDGT utilizes the Homomorphic aggregation method as a SDC algorithm to aggregate the encrypted data without decrypting them when compared to CSDA, iSecLEACH, SM-LEACH, and USCDRP, which reduces the energy consumption and the computation power. In addition, SEEDGT proposed the modified CS method which achieves fair use of CS and removes the unnecessary uses of CS in each cluster even if the cluster does not need to use it. This contradicts the method in SDC that does not efficiently achieves the energy load balance. On the other hand, SEEDGT allows each cluster to dynamically re-elect its new CH without the need to exchange extra information with the BS as others algorithms are doing. In this way, SEEDGT reduces the transmission data, leads to reduce the power consumption, and increases the network lifetime

5.2. Coefficients analysis

In this section, we investigate how the different values of the parameters α, β, γ, $w1$, $w2$, and $w3$ affect the test results. In the first test, we discuss the effect of α, β, and γ, while the effect of $w1$, $w2$, and $w3$ are analyzed in the second test.

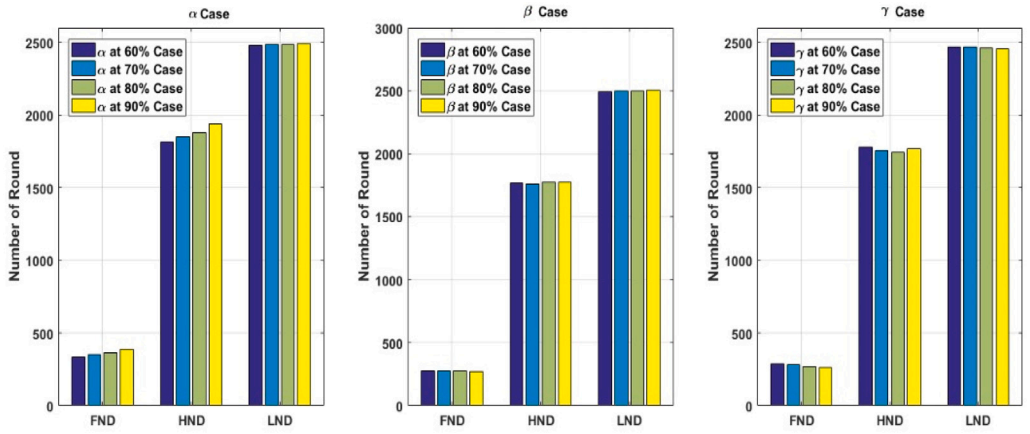

Fig. 6. α, β, and γ analysis.

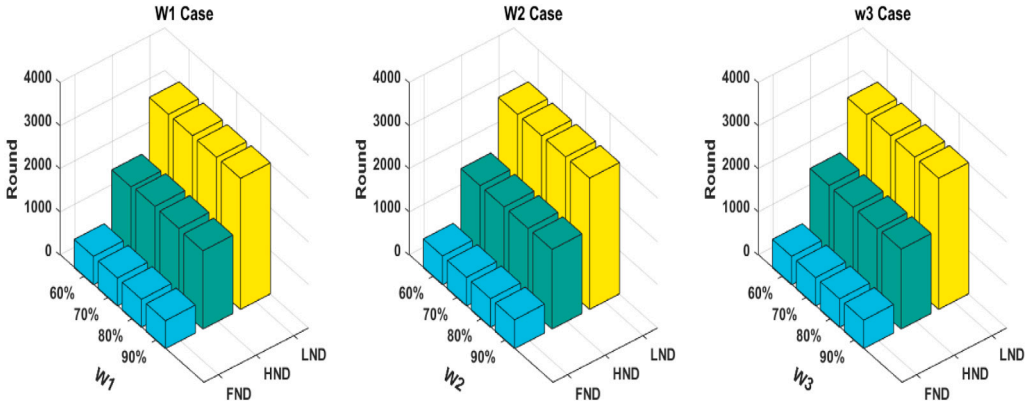

Fig. 7. $w1$, $w2$, and $w3$ analysis.

In the first test, we considered the same setting described previously and we fixed the weights of the trust factors w_1, w_2, and w_3 in Eq. (8). Using Eq. (9), we investigate the effect of varying the values of coefficients α, β, and γ on the network lifetime (FND, HND and LND) by analyzing the effect of the coefficients on energy, distance, and trust feedback factors. The following three different cases are considered αCase, βCase, and γCase, where in each case a coefficient value is maximized compared to the others:

- αCase: we set $\alpha = 0.6, 0.7, 0.8, 0.9$, with $\beta = \gamma = 0.2, 0.15, 0.1, 0.05$.
- βCase: we set $\beta = 0.6, 0.7, 0.8, 0.9$, with $\alpha = \gamma = 0.2, 0.15, 0.1, 0.05$.
- γCase: we set $\gamma = 0.6, 0.7, 0.8, 0.9$, with $\alpha = \beta = 0.2, 0.15, 0.1, 0.05$.

The results of this test are shown in Fig. 6. It is noticed that in αCase, as the α value increases FND, HND, and LND increase. In βCase and γCase, as the γ and β values increase, FND slightly decreases whereas HND and LND increase. This is because α is the coefficient of energy factor and so the probability of selecting a node with low energy to be a CH decreases. As β and γ are the coefficients of distance, and trust feedback factors respectively, the probability of selecting a node with low energy to be a CH increases which leads to nodes with low energy value to work as CHs and nodes with high energy values to work as members. This leads to an increase of HND and LND.

In the second test, we investigate how the different values of parameters $w1$, $w2$, and $w3$ affect the simulation results by discussing the effect of $w1$, $w2$, and $w3$ on SPR, FR, and ETR as in Eq. (8). In this test, we consider the same setting described previously and we fixed the weights of α, β, and γ coefficients in Eq. (9). We consider the following three cases: $w1$Case, $w2$Case, and $w3$Case where in each case, a different coefficient value is maximized compared to the others:

- $w1$Case: we set $w1 = 0.6, 0.7, 0.8, 0.9$, with $w2 = w3 = 0.2, 0.15, 0.1, 0.05$.
- $w2$Case: we set $w2 = 0.6, 0.7, 0.8, 0.9$, with $w1 = w3 = 0.2, 0.15, 0.1, 0.05$.
- $w3$Case: we set $w3 = 0.6, 0.7, 0.8, 0.9$, with $w1 = w2 = 0.2, 0.15, 0.1, 0.05$.

The results of this test are shown in Fig. 7. We can notice that in $w1$Case, $w2$Case, and $w3$Case, there are slight changes in FND, HND, and LND. FND, HND, and LND values slightly decrease as $w1$ increases and slightly increase the values of FND, HND and LND as $w2$ increases, and FND and LND fluctuate with the increase in $w3$. Eq. (8) is employed indirectly by the BS to figure out the feedback value in Eq. (9) for network nodes and Eq. (8) is employed by CH to figure out the trustworthiness of member nodes and gets the total aggregated trust value in Eq. (22). In both cases, the trustworthiness factor is one of the three factors in Eqs. (9), (22) and it is affected by γ coefficient. As a result, slight changes occur in FND, HND, and LND as the values of $w1$, $w2$, and $w3$ change.

5.3. Evaluation of TEECA algorithm

In this part, we analyze our TEECA using the same steady phase as in previous clustering algorithms (e.g., EESTBCA Rehman et al. (2017), HiTSeC Gaber et al. (2018), Trusted-LEACH Narayan et al. (2019) and LEACH-TM Fang et al. (2021)), where data acquired from nodes is sent to CHs and subsequently transferred to BS. Besides, we consider the reconfiguration phase along with TEECA during this test.

The simulations in MATLAB R2015a consider 100 nodes randomly deployed in a 2D area 100×100 m^2, with the BS at the center. The

Table 2
Simulation parameters.

Parameters	Values
Network area	100 m × 100 m
E_{elec}	50 nJ/bit
E_{DA}	5 nJ/bit/signal
Packet size	4000 bits
ϵ_{mp}	0.00013 pJ/bit/m^4
Node's initial energy	random (with total 102 J)
d_0	87 m
ϵ_{fs}	10 pJ/bit/m^2

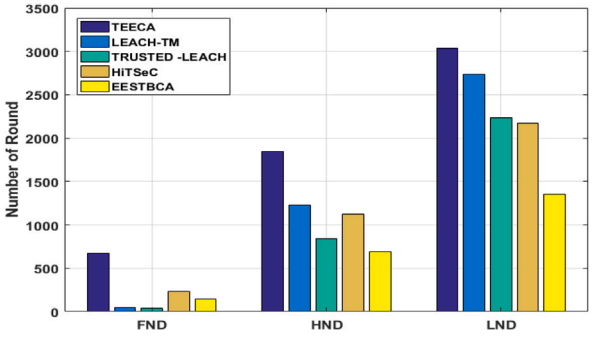

Fig. 8. FND, HND and LND in TEECA, EESTBCA, HiTSeC, Trusted-LEACH and LEACH-TM algorithms.

system model and parameters used can be seen in Section 4.1, and the strategy is evaluated using several topologies created at random. The total energy of the WSN is assumed to be 102 J. Used parameters in simulation can be found in Table 2.

For evaluation the following performance metrics are used :

1. Network lifetime (First (FND), Half (HND) and Last node (LND) dead).
2. Average residual energy per round: the total residual energy of all nodes divided by the total number of nodes.
3. Alive nodes count per round: number of nodes that are alive in WSN per round.
4. Throughput (Rodríguez et al., 2020): The number of the sent packets to the BS.

In this test, 100 nodes are considered. Based on the above metrics, we analyze TEECA with: EESTBCA (Rehman et al., 2017), HiTSeC (Gaber et al., 2018), Trusted-LEACH (Narayan et al., 2019) and LEACH-TM (Fang et al., 2021).

Fig. 8 depicts the lifetime of the network (FND, HND and LND respectively). Fig. 9 shows the number of alive nodes per round in TEECA, EESTBCA, HiTSeC, Trusted-LEACH and LEACH-TM algorithms. It is evident that the count of alive nodes per round in TEECA is higher than that of EESTBCA, HiTSeC, Trusted-LEACH and LEACH-TM algorithms. From Figs. 8 and 9, it is evident that TEECA improves the lifetime of WSN in terms of the FND compared with EESTBCA, HiTSeC, Trusted-LEACH and LEACH-TM algorithms because of the TEECA's efficiency in cluster creation and selection of CH, which enables each CM to send data with minimal energy usage.

The average of energy consumed by the nodes per round with the increase in rounds are presented in Fig. 10. The results in these are taken considering only one topology to show the distribution of energy during operational rounds. We can notice that the consumed energy in each round by TEECA is less an consumed energy by EESTBCA, HiTSeC, Trusted-LEACH and LEACH-TM algorithms. That is because, during the reconfiguration phase of TEECA, the CHs are selected dynamically inside each cluster without sharing a lot of extra information with the BS. This reduces the communication cost for TEECA and reduces the power consumption.

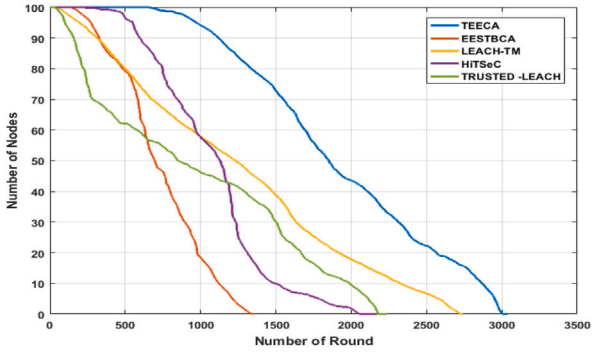

Fig. 9. Alive nodes count per round in TEECA, EESTBCA, HiTSeC, Trusted-LEACH and LEACH-TM algorithms.

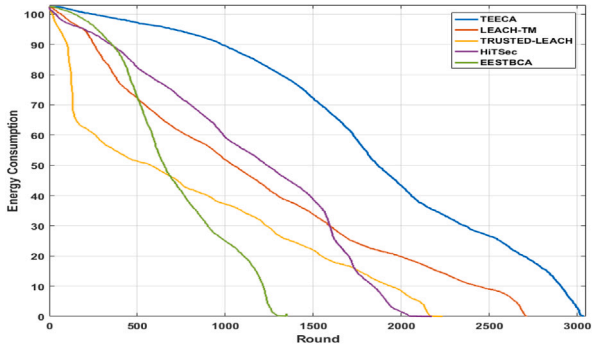

Fig. 10. The energy per round in TEECA, EESTBCA, HiTSeC, Trusted-LEACH and LEACH-TM algorithms.

Next, we provide the comparison of total number of packets to BS under different percentage of malicious nodes: 15%, 30%, and 45%. The results are illustrated in Figs. 11(a), 11(b) and 11(c). In this test, the nodes count varies from 50 to 300 in increment of 50. We can notice the increase in the number of data packets sent to the BS by the proposed scheme than the others. As can be observed, the implementation of trust management scheme can successfully detect malicious nodes and prevent data packet loss. This is the advantage of the proposed trust management system in which the nodes that exhibit malicious behavior are not permitted to be selected as CH.

5.4. Effect of transmission radius

In this test we discuss the effect of Transmission Radius (TR) on SEEDGT in terms of network lifetime, and energy usage, assuming that the nodes have different values for TR. This test is executed in the following environment: A network of size 100 m × 100 m, with 250 deployed nodes and the BS in the middle. TR values vary from 15 m to 60 m with a 15 m increment. For radio communication, we apply the same energy dispersion model as in Section 4.1.

As communication distance increases, single hop communication requires more energy, becoming a less energy efficient approach and causing uneven energy levels at the nodes. When the TR is increased, the neighbor nodes count is increased, which increases the probability of the cluster size being increased. The use of CS method helps to work effectively in this regard. Fig. 12 shows that SEEDGT exhibits enhanced performance in terms of network lifetime with and without CS, because the overall transmitted data size is lowered using the CS approach, which is regarded to be the key factor impacting the node's power consumption. Also, Fig. 13 shows the SEEDGT residual energy performance with and without CS. That is because, according to the proposed modified plain CS approach, only clusters with a size more than or equal to M would compress their data; otherwise,

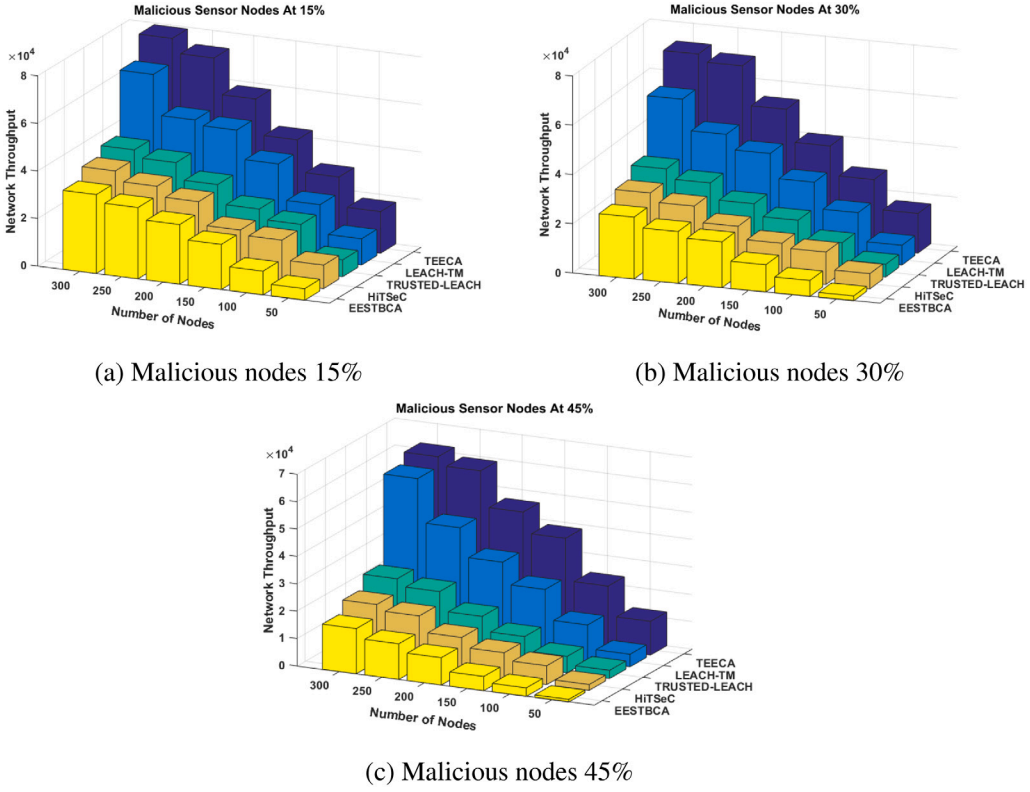

Fig. 11. Comparison of total number of packets to BS under malicious nodes is 15%, 30%, and 45%.

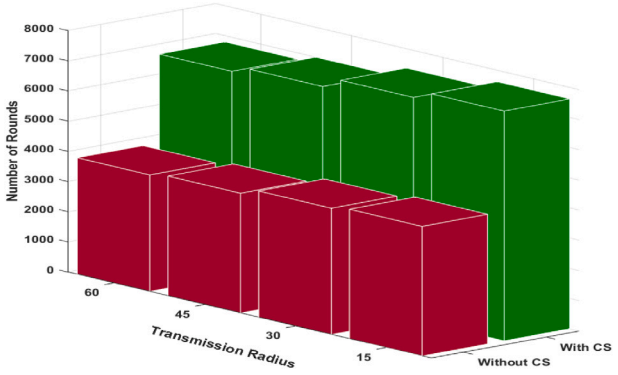

Fig. 12. Network lifetime till half of nodes die as a function of TR values in SEEDGT when CS is employed and without CS (no compression).

the data will be sent without compression, reducing the amount of unnecessary compressed data and achieving load balance across sensor nodes. Thus, SEEDGT succeeds in achieving the balance between CS and an encryption algorithm, and offers better performance than using the encryption algorithm without CS.

6. Conclusion

In this paper, we have introduced an integrated technique based on trust mechanism, public key algorithm and CS. To improve security and energy efficiency, the proposed technique provides data compression and secure data aggregation. The proposed technique operates in three main phases, namely cluster formation, network operation and reconfiguration phase. During the Cluster formation phase, energy-aware and trust based methods are applied for the creation of clusters and cluster head selection. Network operation phase has three steps with BS generating public and private keys in the first stage and then transmitting a public key to the entire network for data encryption. During the second stage, the sensors send the cipher data to the CHs that use the CS matrix for data compression and aggregation, and then send the M samples to the BS if required otherwise, send cipher data without applying CS (if it is less than M). This helps to eliminate unnecessary traffic. Finally, the BS retrieves the ciphered data and decrypts it. It then reconstructs the compressed samples in order to acquire the original data from the received CS data. The changes that could occur during network operation is considered in the Reconfiguration phase. The simulation results demonstrate that the proposed technique is effective in improving the security and network lifetime of the IoT-WSN network.

CRediT authorship contribution statement

Ahmed Salim: Conceptualization, Validation, Formal analysis, Resources, Data curation, Visualization, Funding acquisition, Investigation. **Walid Osamy:** Conceptualization, Methodology, Investigation, Data curation, Writing – original draft, Project administration, Formal analysis. **Ahmed Aziz:** Conceptualization, Visualization, Investigation, Writing – original draft, Methodology, Formal analysis, Data curation. **Ahmed M. Khedr:** Conceptualization, Methodology, Supervision, Formal analysis, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors gratefully acknowledge Qassim University, Saudi Arabia, represented by the Deanship of Scientific Research, on the financial support for this research under the number (mcs- as-2020-1-3-I-10160) during the academic year 1441 AH/2020 AD.

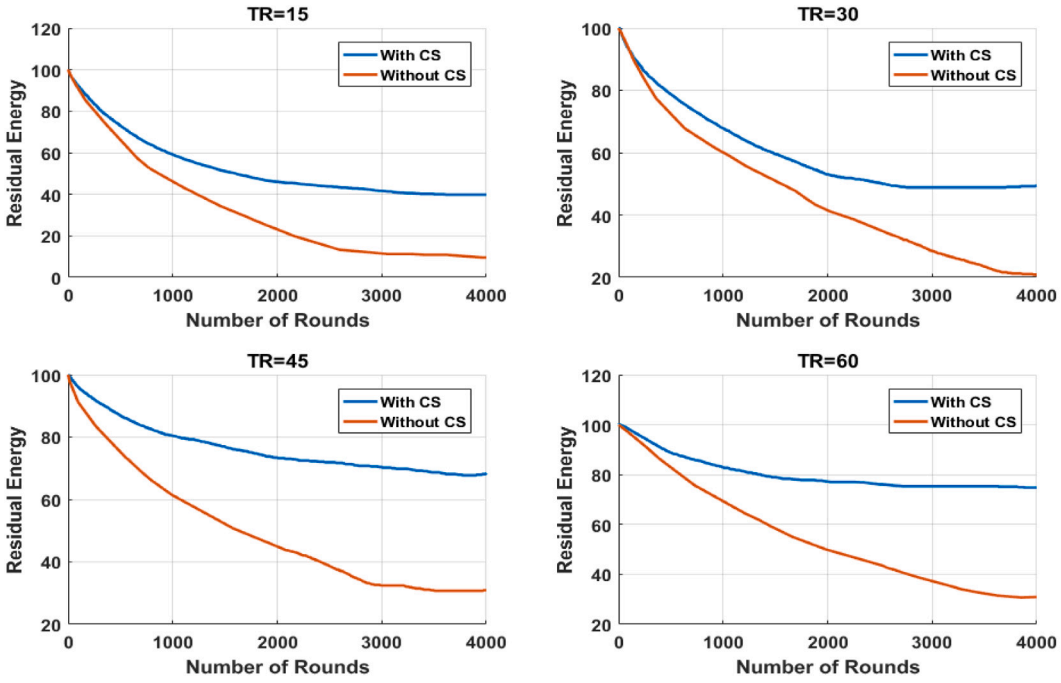

Fig. 13. Residual energy as a function of TR values in SEEDGT with CS and without CS (non-compression).

References

Abdallah, W, Boudriga, N, Kim, D, An, S., 2015. An efficient and scalable key management mechanism for wireless sensor networks. In: 2015 17th International Conference on Advanced Communication Technology (ICACT).

Abu Salem, A.O, Shudifat, N., 2019. Enhanced LEACH protocol for increasing a lifetime of WSNs. Pers. Ubiquit. Comput. 23, 901–907. http://dx.doi.org/10.1007/s00779-019-01205-4.

Afsar Mehdi, M., Mohammad-H, ., Tayarani-N, ., 2014. Clustering in sensor networks: A literature survey. J. Netw. Comput. Appl. 46, 198–226.

aghavaraju, D., 2017. Secure data communication in I-Leach protocol in wireless sensor networks. I-Manager's J. Wirel. Commun. Netw. 6 (1), 7.

Aziz, Ahmed, Osamy, Walid, Khedr, Ahmed M., El-Sawy, Ahmed A., Singh, Karan, 2020a. Grey wolf based compressive sensing scheme for data gathering in IoT based heterogeneous WSNs. Wirel. Netw. 1–24.

Aziz, Ahmed, Osamy, Walid, Khedr, Ahmed M., Salim, Ahmed, 2021. Chain-routing scheme with compressive sensing-based data acquisition for Internet of Things-based wireless sensor networks. IET Netw. 10 (2), 43–58.

Aziz, Ahmed, Singh, Karan, Osamy, Walid, Khedr, Ahmed M., 2019. Effective algorithm for optimizing compressive sensing in IoT and periodic monitoring applications. J. Netw. Comput. Appl. 126, 12–28.

Aziz, Ahmed, Singh, Karan, Osamy, Walid, Khedr, Ahmed M., 2020b. An efficient compressive sensing routing scheme for Internet of Things based wireless sensor networks. Wirel. Pers. Commun. 114, 1905–1925.

Castillejo, P, Martínez-Ortega, J.-F, López, L, Alcón, J.A.S., 2015. SensoTrust: TRust-worthy domains in wireless sensor networks. Int. J. Distrib. Sens. Netw. 11 (7), 484820.

Chatterjee, M, Das, S.K, Turgut, D., 2002. WCA: A Weighted clustering algorithm for mobile ad hoc networks. Cluster Comput. 5 (2), 193–204.

Chong, L, Feng, W, Jun, S, Chang, C., 2009. Compressive data gathering for large-scale wireless sensor networks, In: Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, MobiCom '09, 145–156, New York, NY, USA.

Dhillon, P.K, Kalra, S., 2016. Elliptic curve cryptography for real time embedded systems in IoT networks. In: 5th International Conference on Wireless Networks and Embedded Systems (WECON). Rajpura, pp. 1–6.

Dhulipala, V.R.S, Karthik, N., 2017. Trust management technique in wireless sensor networks: challenges and issues for reliable communication: a review. CSIT 5, 281–294. http://dx.doi.org/10.1007/s40012-017-0169-5.

Diffie, W, Hellman, M.E., 1976. New directions in cryptography. IEEE Trans. Inform. Theory 22, 644–654.

Fang, Wei, Wen, XueZhi, Xu, Jiang, Zhu, JieZhong, 2019. CSDA: A novel cluster-based secure data aggregation scheme for WSNs. Cluster Comput. 22 (3), 5233–5244.

Fang, Weidong, Zhang, Wuxiong, Chen, Wei, Pan, Tao, Ni, Yepeng, Yang, Yinxuan, 2020. Trust-based attack and defense in wireless sensor networks: A survey. Wirel. Commun. Mob. Comput. 2020.

Fang, Weidong, Zhang, Wuxiong, Yang, Wei, Li, Zhannan, Gao, Weiwei, Yang, Yinxuan, 2021. Trust management-based and energy efficient hierarchical routing protocol in wireless sensor networks. Digit. Commun. Netw..

Feng, R, Xu, X, Zhou, X, Wan, J., 2011. A trust evaluation algorithm for wireless sensor networks based on node behaviors and D-S evidence theory. Sensors 11 (2), 1345–1360.

Gaber, Tarek, Abdelwahab, Sarah, Elhoseny, Mohamed, Hassanien, Aboul Ella, 2018. Trust-based secure clustering in WSN-based intelligent transportation systems. Comput. Netw. (ISSN: 1389-1286) 146, 151–158. http://dx.doi.org/10.1016/j.comnet.2018.09.015.

Gentry, C, Sahai, A, Waters, B., 2013. Homomorphic encryption from learning with errors:conceptually-simpler, asymptotically-faster, attribute-based. In: Advances in Cryptography. Springer, Berlin, pp. 75–92.

Ghaderi, M, Vakili, Vahid.Tabataba, Sheikhan, Mansour., 2020. FGAF-CDG: Fuzzy geographic routing protocol based on compressive data gathering in wireless sensor networks. J. Ambient Intell. Humaniz. Comput. 11 (6), 2567–2589.

Gulen, U, Alkhodary, A, Baktir, S., 2019. Implementing RSA for wireless sensor nodes. Sensors 19 (13), 28–64.

Hayouni, Haythem, Hamdi, Mohamed, 2017. A data aggregation security enhancing scheme in WSNs using homomorphic encryption. Intell. Autom. Soft Comput. 1–9.

Hsieh, Sung-Hsien, Hung, Tsung-Hsuan, Lu, Chun-Shien, Chen, Yu-Chi, Pei, Soo-Chang, 2018. A secure compressive sensing-based data gathering system via cloud assistance. IEEE Access 6, 31840–31853.

Ifzarne, Samir, Hafidi, Imad, Idrissi, Nadia, 2020. Compressive Sensing Based on Homomorphic Encryption and Attack Classification using Machine Learning Algorithm in WSN Security. In: Proceedings of the 3rd International Conference on Networking, Information Systems and Security, pp. 1–6.

Ifzarne, Samir, Hafidi, Imad, Idrissi, Nadia., 2021. Secure data collection for wireless sensor network. In: Emerging Trends in ICT for Sustainable Development. Springer, Cham, pp. 241–248.

Karl, H, Willig, A., 2007. Protocols and Architectures for Wireless Sensor Networks. Wiley, England.

Khediri, S.E, Nasri, N, Wei, A, Kachouri, A., 2014. A new approach for clustering in wireless sensors networks based on LEACH. Procedia Comput. Sci. 32, 1180–1185.

Khedr, A.M., 2105. Effective Data Acquisition Protocol for Multi-hop Heterogeneous Wireless Sensor Networks Using Compressive Sensing, Algorithms, 8 (4) 910-928. http://dx.doi.org/10.3390/a8040910.

Kodali, RK, Soratkal, SR., 2015. Trust model for WSN. In: 2015 International Conference on Applied and Theoretical Computing and Communication Technology (ICATccT). IEEE.

Kumar, Manish, Verma, Shekhar, Lata, Kusum., 2015. Secure data aggregation in wireless sensor networks using homomorphic encryption. Int. J. Electron. 102 (4), 690–702.

Lalama, Zahia, Boulfekhar, Samra, Semechedine, Fouzi, 2021. Localization optimization in WSNs using meta-heuristics optimization algorithms: A survey. Wirel. Pers. Commun. 1–24.

Li, Xiaoyong, Zhou, Feng, Du, Junping, 2013. LDTS: A Lightweight and dependable trust system for clustered wireless sensor networks. IEEE Trans. Inf. Forensics Secur. 8 (6), 924–935.
Liu, Zhen, Han, Yi-Liang, Yang, Xiao-Yuan, 2019. A compressive sensing–based adaptable secure data collection scheme for distributed wireless sensor networks. Int. J. Distrib. Sens. Netw. 15 (6), 1550147719856516.
Lv, C, Wang, Q, Yan, W, Li, Jia, 2019. Compressive sensing-based sequential data gathering in WSNs. Comput. Netw. 154, 47–59.
Mishra, Mukesh, Gupta, Gourab Sen, Gui, Xiang, 2019. Trust-based cluster head selection using the K-means algorithm for wireless sensor networks. In: 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE, pp. 819–825.
Narayan, Vipul, Daniel, A.K., 2021. A novel approach for cluster head selection using trust function in WSN. Scalable Comput.: Pract. Exp. 22 (1), 1–13.
Narayan, ., Deena, B., Vineetha, P, Kumar Raju Alluri, BKSP., 2019. Enhanced trust-based cluster head selection in wireless sensor networks. In: Innovations in Computer Science and Engineering. Springer, Singapore, pp. 263–275.
Needell, D, Tropp, J.A., 2009. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmon. Anal. 26 (3), 301–321.
Osamy, W, El-sawy, A.A, Khedr, A.M., 2019a. SATC: A Simulated annealing based tree construction and scheduling algorithm for minimizing aggregation time in wireless sensor networks. Wirel. Pers. Commun. http://dx.doi.org/10.1007/s11277-019-06440-9.
Osamy, W, Khedr, A.M., 2020. Adaptive and dynamic mechanism for round length determination in cluster based wireless sensor networks. Wirel. Pers. Commun. http://dx.doi.org/10.1007/s11277-020-07413-z.
Osamy, W, Khedr, A.M, Aziz, A, El-Sawy, A., 2019b. Cluster-tree routing scheme for data gathering in periodic monitoring applications. IEEE Access 6 (Page(s)), 77372–77387.
Pacharaney, U.S, Gupta, R.K., 2019. Cluster restructuring and compressive data gathering for transmission efficient wireless sensor network. In: Intelligent Communication Technologies and Virtual Mobile Networks. Springer, Cham, pp. 1–18.
Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology - EUROCRYPT'99. Springer, Berlin, pp. 223–238.
Priyadharshini, ., Sujatha, A., Arvind, C., 2021. Security-based LEACH protocol for wireless sensor network. In: International Conference on Innovative Computing and Communications. Springer, Singapore, pp. 855–863.
Ramalingam, L, Audithan, S., 2014. Trust based cluster head selection algorithm for wireless sensor network. In: 2014 2nd International Conference on Current Trends in Engineering and Technology (ICCTET). IEEE.
Rehman, Eid, Sher, Muhammad, Abbas Naqvi, Syed Hussnain, Khan, Khan Badar, Ullah, Kamran, 2017. Energy efficient secure trust based clustering algorithm for mobile wireless sensor network. J. Comput. Netw. Commun. 2017.
Rivest, R.L, Adleman, L, Dertouzos, M.L., 1978. On data banks and privacy homomorphisms. In: Foundation of Secure Computation. Academic Press, New York, pp. 169–179.
Rodríguez, Alma, Del-Valle-Soto, Carolina, Velázquez, Ramiro, 2020. Energy-efficient clustering routing protocol for wireless sensor networks based on yellow saddle goatfish algorithm. Mathematics 8 (9), 1515.
Sabale, Ketan, Mini, S., 2021. Localization in wireless sensor networks with mobile anchor node path planning mechanism. Inform. Sci. 579, 648–666.
Saidi, Ahmed, Benahmed, Khelifa, Seddiki, Nouredin, 2020. Secure cluster head election algorithm and misbehavior detection approach based on trust management technique for clustered wireless sensor networks. Ad Hoc Netw. 106, 102215.
Salehi, Mohsen, Karimian, Jamal, 2017. A trust-based security approach in hierarchical wireless sensor networks. Ad Hoc Netw. 7 (6), 58–67.
Salim, Ahmed, Osamy, Walid, Khedr, Ahmed M., Aziz, Ahmed, Abdel-Mageed, M., 2020. A secure data gathering scheme based on properties of primes and compressive sensing for IoT based WSNs. IEEE Sens. J..
Samydurai, Gopinath, Bojan, Gurumoorthy.Kambatty, Dharmarajan, Bhanu., 2020. Fuzzy based secure data gathering approach for ad hoc sensor networks. J. Sci. Ind. Res. (JSIR) 79 (05), 391–394.
Selvi, M, Thangaramya, K., Ganapathy, Sannasi, Kulothungan, Kanagasabai, Nehemiah, H. Khannah, Kannan, Arputharaj, 2019. An energy aware trust based secure routing algorithm for effective communication in wireless sensor networks. Wirel. Pers. Commun. 105 (4), 1475–1490.
Silmi, Souhila, Doukha, Zouina, Moussaoui, Samira, 2021. A self-localization range free protocol for wireless sensor networks. Peer-to-Peer Netw. Appl. 1–11.
Singh, A, Awasthi, K, Singh, K., 2017. Cryptanalysis and improvement in user authentication and key agreement scheme for wireless sensor network. Wirel. Pers. Commun. 94, 1881–1898.
Smaragdakis, G, Matta, I, Bestavros, A., 2004. SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks. In: Proceeding of the International Workshop on SANPA.
Sun, Z, Xing, X, Song, Bin, Nie, Yalin, Shao, Hongxiang, 2019. Mobile intelligent computing in Internet of Things: An optimized data gathering method based on compressive sensing. IEEE Access 7, 66110–66122.
Tang, Li, Hu, Haibo, 2020. OHEA: Secure Data Aggregation in Wireless Sensor Networks against Untrusted Sensors. In: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, pp. 1425–1434.
Tirani, S, Pakdaman, S, Avokh, A, Azar, S., 2020. WDAT-OMS: A Two-level scheme for efficient data gathering in mobile-sink wireless sensor networks using compressive sensing theory. IET Commun. 14 (11), 1826–1837.
Tropp, J, Gilber, A., 2007. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inform. Theory 53 (14), 4655–4666.
Vijayalakshmi, V, Senthilkumar, A., 2020. USCDRP: Unequal secure cluster-based distributed routing protocol for wireless sensor networks. J. Supercomput. 76 (2), 989–1004.
Wang, Q, Lin, D, Yang, P, Zhang, Z., 2019a. An energy-efficient compressive sensing-based clustering routing protocol for WSNs. IEEE Sens. J. 19 (10), 3950–3960, 15 May15.
Wang, M, Xiao, D, Ao, Z., 2019b. A novel privacy-preserving data gathering scheme in WSN based on compressive sensing and embedding. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, pp. 1–6.
Wei, D, Olgica, M., 2009. Subspace pursuit for compressive sensing signal reconstruction. IEEE Trans. Inf. Theory 55 (5), 2230–2249.
Wei, Zhe, Yu, Shuyan, 2018. Energy aware and trust based cluster head selection for ad-hoc sensor networks. IJ Netw. Secur. 20 (3), 496–501.
Xiaohan, Y.U., Keming, D.O.N.G., Xia, L.I., Chao, Chen, 2019. A compressive data gathering method based on ElGamal cryptography. Telecommun. Sci. 35 (12), 67.
Xu, Y, Sun, G, Geng, T, Zheng, B., 2019. Compressive sparse data gathering with low-rank and total variation in wireless sensor networks. IEEE Access 7, 155242–155250.
Yan, Zheng, Zhang, Peng, Vasilakos, Athanasios V., 2014. A survey on trust management for internet of things. J. Netw. Comput. Appl. 42, 120–134.
Yu, X, Xu, W, Chen, C., 2020. HDCDS-CDG: A Hierarchically diffused connected dominating sets based compressed data gathering scheme. J. Phys. Conf. Ser. 1570 (1), 012056, IOP Publishing.
Zhang, C, Li, Ou, Tong, Xin, Ke, Ke, Li, Mingxuan, 2019. Spatiotemporal data gathering based on compressive sensing in wsns. IEEE Wirel. Commun. Lett. 8 (4), 1252–1255.
Zhang, Qiang, Liu, Xiaowu, Yu, Jiguo, Qi, Xiaohan, 2020. A trust-based dynamic slicing mechanism for wireless sensor networks. Procedia Comput. Sci. 174, 572–577.
Zhang, Ping, Wang, Shaokai, Guo, Kehua, Wang, Jianxin, 2018a. A secure data collection scheme based on compressive sensing in wireless sensor networks. Ad Hoc Netw. 70, 73–84.
Zhang, Ping, Wang, Jianxin, Guo, Kehua, Wu, Fan, Min, Geyong, 2018b. Multi-functional secure data aggregation schemes for WSNs. Ad Hoc Netw. 69, 86–99.

Ahmed Salim received the B.Sc. degree in computer science and the M.Sc. degree in distributed computing from Zagazig University, Egypt, in 2001 and 2006, respectively, and the Ph.D. degree in systems, network, and telecommunication devices from the Bonch-Bruvich University of Telecommunication, Saint-Petersburg, Russia, in 2010. In 2012, he was a Consultant of the Information and Communication Technology Project (ICTP) at Zagazig University. From 2011 to 2019, he was an Assistant Professor with the Department of Mathematics, Faculty of Science, Zagazig University, where he has been an Associate Professor, since 2019. Since 2014, he has been an Assistant Professor with the Department of Mathematics, Faculty of Science and Art, Qassim University, Al-Mithnab, Saudi Arabia. His research interests include decomposable algorithms, computing, the IoT, and wireless sensor networks

Walid Osamy received his B.Sc. degree with honor in computer science in June 2000, the M.S. degree in June 2006, and the PhD. Degree in September 2010 all in area of computer science and from the Faculty of science, Zagazig University, Egypt. He has been involved in the projects of network infrastructure, and management information systems with the Communication Information Technology Center (CITC), Zagazig University, Egypt. From 2010 to 2019, he has been an Assistant Professor with the Department of Computer Science, Faculty of Computers and Informatics, Benha University, Egypt. Since 2019, he has been an Associate Professor with the Department of Computer Science, Faculty of Computers and Artificial Intelligence, Benha University. Since 2015, he has been an Assistant Professor with Qassim University, Buridah, Saudi Arabia. His research interests include computational intelligence and in the field of IoT (mobile computing and wireless sensor networks (WSNs)).

Ahmed Aziz received Ph.D. from school of computer and system science, Jawaharlal Nehru University, New Delhi, India, his M.S. degree in OCT 2014, in area of computer science from the Faculty of Computers and Informatics, Benha University, Benha, Egypt, and his B.Sc. degree with honor in Computer science in June 2007. From December 2007 till 2010 he has been working at Faculty of Science, Benha University, Benha, Egypt, and from 2010 till now he has been at Computer Science Department, Faculty of computers and Artificial Intelligence , Benha University, Benha, Egypt (www.fci.bu.edu.eg). His research interests include sensor networks, Compressive sensing, computing, wireless networks and IoT.

Ahmed M. Khedr received his B.Sc degree in Mathematics in June 1989 and the M.Sc degree in the area of optimal control in July 1995, both from Zagazig University, Egypt. In July 1999, he received his M.Sc and in March 2003, he received his Ph.D. degrees, both in Computer Science and Engineering, from University of Cincinnati, Ohio, USA. From March 2003 to January 2004, he was a research assistant professor at ECECS Department University of Cincinnati, USA. From January 2004 to May 2009, he worked as an assistant professor at Zagazig University, Egypt. From September 2009 to September 2010 he worked as an associate professor at the Department of Computer Science, College of Computers and Information Systems, Taif University, KSA. Since December 2014, he is a Professor at Zagazig University, Egypt. From September 2010 till December 2019, he worked as an Associate Professor and since January 2020, is a Professor at the Department of Computer Science, College of Computing and Informatics, University of Sharjah, UAE. He was awarded the State Prize of distinction in advanced technology, Sharjah Islamic Bank prize of distinction in research and the University of Sharjah prize of distinction in research, in June 2009, May 2013 and April 2014, respectively. His research interests include Wireless Sensor Networks, Internet of Things, and Distributing Computing.