Energy Efficient Key for Heterogeneous WSN- IoT

Thirupathi Durgam[1] **Dr. Ritesh Sadiwala**[2]
Assistant Professor,Dept. of ECE, St.Martins Engineering College,Telangana
Professor, Dept. of ECE,RKDF University,Bhopal

Abstract:*Wireless Sensor Networks (WSNs) consist of lightweight devices to measure sensitive data that are highly vulnerable to security attacks due to their constrained resources. In a similar manner, the internet-based lightweight devices used in the Internet of Things (IoT) are facing severe security and privacy issues because of the direct accessibility of devices due to their connection to the internet. Since there can be enormous number of devices with different functions connected to IoT network, each devices may communicate with uncertain number of devices. Some messages sometimes should be delivered to multiple devices. For the purpose of efficiency and communication performance, the group communication can be employed in the network when there is a necessity in sending messages to several recipients. This paper, introduces a new heterogeneous-aware routing protocol well known as Modified zonal-stable election protocol (MZ-SEP) Routing Protocol with Hierarchical Clustering Approach for Wireless Heterogeneous Sensor Network or MZ-SEP, where the connection of nodes to a base station (BS) is done via a hybrid method, i.e., a certain amount of nodes communicate with the base station directly, while the remaining ones form a cluster to transfer data.However, to ensure a safe IoT environment by proposing an efficient key management technique that uses a combination of symmetric and asymmetric cryptosystem to obtain the speed of the former as well as the security benefits of the latter. Our proposal considers a set of Smart Objects (SO) capable of key registration, generation and distribution for IoT data transmission.The suitability of the proposed approach is measured experimentally and the results are comparable to existing works with respect to key conversion time, algorithm execution time, and bandwidth utilization.*

Keywords: *Internet of things; smart objects; key management;hashing, Z-SEP*

I. INTRODUCTION

Wiresless sensor networks (WSNs) are rapidly growing in popularity due to the low-cost solutions for a variety of challenges in the real-world. WSN has no infrastructure support, is quickly deployed in a region with several low-cost sensor nodes, is employed for monitoring the environment, and is rigid to maintain its A recent advancement in communication technology, Wireless Sensor Networks (WSNs) are widely used in several applications [**1**]. The science community has focused on the security of WSNs. Because of the resource-constrained environment of the WSNs, classic security mechanisms are not practical since they consume too much energy; hence researchers are proposing new lightweight security mechanisms for every possible security aspect of WSNs. WSNs consist of many small, low-cost, self-governing ends called sensor nodes with little ability to manipulate data [**2**] and with constrained computing, energy, and memory. In a heterogeneously grouped approach, as described in Figure 1, the low end sensor (L-sensors) are resource-constrained devices with low power, short communication range, limited memory, and less computation power. On the other hand, H-sensors are equipped with tamper resistance and have enough resources, like high battery power, broad communication ranges, sufficient memory, and high computational capabilities. L-sensors are deliberately conveyed in a group, and every group is controlled by a group head (H-sensor). The L-sensors essentially sense environmental statistics and forward it to the H-

sensors and the other way around. H-sensors can perform complex operations on the sensor information, and utilize longer radio and straightforwardly transfer it to the base-station. The base-station (BS) is a powerful hub, and it has abundant sources. The BS might be a remote server, and it might be connected with the external world utilizing the accelerated Internet.

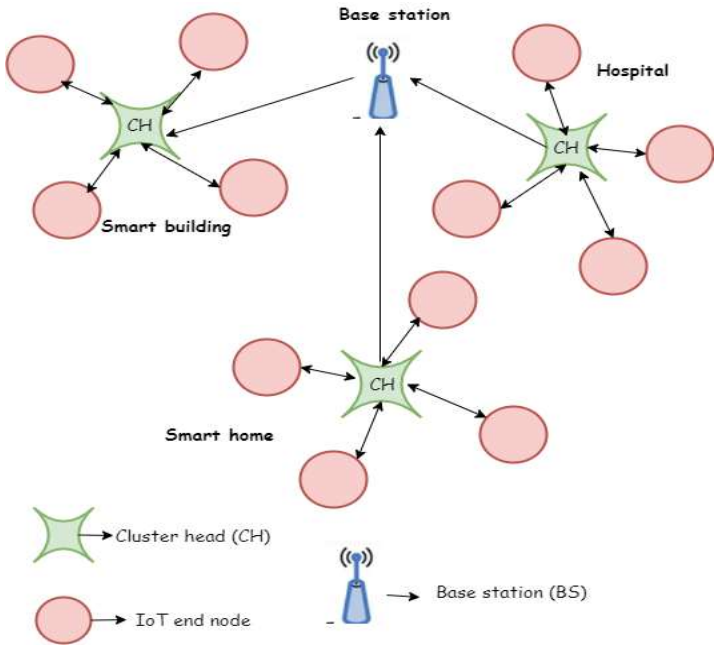

Figure 1: Heterogeneous WSN-based IoT architecture

Before the nodes' deployment in WSNs, each sensor node is pre-configured with a set of symmetric keys shared with all the other sensor nodes of the network to transmit IDs securely. After the network deployment, every sensor node identifies the specified symmetric keys used to communicate with the cluster head (CH) [**3**]. The cluster-head maintains all the symmetric keys shared with the sensor nodes belonging to its cluster. The main reason the use of these symmetric keys is to facilitate the multi-hop communication while transmitting secret data, particularly, the personal-proportion distribution and the exchanged facts among the cluster-heads and the base station (BS). The implementation of the polynomial that is applied for deriving an intra-group key can reduce the session key storage overhead at the member nodes and their CH. After the intra-group key is acquired, the member nodes self-generate the polynomial functions, which are necessary for growing an inter-organization key. This facilitates the reduction of the communication overhead on the CH. We have proposed a key generation and update mechanism for secure data sharing in identical clusters and among different cluster.

Some of these techniques utilize asymmetric cryptosystems while others utilize symmetric key cryptosystems. Asymmetric-key cryptosystems typically provide more security compared to symmetric-key cryptosystems but it still suffers from high computation overhead [**4, 5**]. In our proposed approach, we use a combination of symmetric and asymmetric cryptosystem that utilizes the speed of the former as well as the benefits of the latter. Usually, in an IoT environment, various sensors are connected with a node (i.e., home appliances in case of a smart home) while that node may be connected with other nodes. If we consider some of the nodes as Smart Object (SO) and the entire sensor data are processed through these nodes, a new communication model can be deduced. We define a

smart object as an entity that has more computational capabilities and can perform various tasks, such as authentication, storage management etc. We found that using the power of symmetric-key cryptosystems with SO can help us solve the existing problems in key management in the IoT environment. In this paper, our contribution is twofold. First, we propose a Smart Object (SO)-based reliable key management technique that defines the key registration, generation, and distribution processes for secure communication between IoT nodes. Second, we define and develop a customized IoT Simulator that can demonstrate the practical usages of the proposed technique by integrating physical SO with simulation data.

II. BACKGROUND WORKS

In this sub-section existing literature that focused on secure key management issues and techniques in WSN with IoT.

In [**6**], the authors witnessed that the security is a critical and vital task in wireless sensor networks, therefore different key management systems have been proposed, many of which are based on symmetric cryptography. Such systems are very energy efficient, but they lack some other desirable characteristics. On the other hand, systems based on public key cryptography have those desirable characteristics, but they consume more energy. Recently based on authenticated messages from base station a new PKC based key agreement protocol was proposed. They show method is susceptible to a form of denial of service attack where resources of the network can be exhausted with bogus messages. Then, proposed two different improvements to solve this vulnerability and corresponding simulation results show that these new protocols retain desirable characteristics of the basic method and solve its deficiencies.

In [**7**], the authors presented energy efficient method based on public key cryptography (PKC) was proposed. They analyzed this protocol and show that it is vulnerable to denial of service (DOS) attacks and adversary can exhaust memory and battery of nodes. Then, analyzed this protocol and show that using a more knowledgeable BS this vulnerability can be solved very efficiently. Based on this observation proposal of a modified version of the protocol that achieves immediate authentication and can prevent DOS attacks. They showed that the improved protocol achieves immediate authentication at the expense of 1.82 mj extra energy consumption while retaining other desirable characteristics of the basic method.

In [**8**] Cloud computing has become an emerging model of information technology industry as it can be accessed anywhere in the world on a pay-per-use basis. However, one major problem it is facing in today's challenging world is the security issues. The various existing key generation and management systems are suffering with the security issues. Along with the security feature, the data sharing is also more important issue to be studied. Considering these two parameters, it is necessary to work on the secure data sharing in public clouds. For that purpose, we have proposed an advanced key management system (AKMS) which can perform identity-based encryption. Also, the proposed AKMS type encryption provides high security, scalability, and flexibility in the public clouds.

In [**9**], the authors focus to develop a secure and flexible multi-factor authentication to telematic environments with key management. To provide security in the use of remote cryptographic functions, they proposed an authentication service embedded in a key management system as a trusted third party. It us evident that the main characteristics of the proposed model are: flexibility, interoperability, safety and mobility. Also, they exposed the model architecture, detail the implementation and analyze the security based on the most common authentication attacks.

In [**10**], the authors focused on key management between mesh and sensor networks. They proposed an efficient key pre-distribution scheme based on two polynomials in wireless mesh networks by employing the nature of heterogeneity. Our scheme realizes the property of bloom filters, i.e., neighbor nodes can discover their shared keys but have no knowledge on

the different keys possessed by the other node, without the probability of false positive. The analysis presented in this paper shows that our scheme has the ability to establish three different security level keys and achieves the property of self-adaptive security for sensor networks with acceptable computation and communication consumption.

In [**11**], the authors focused on secure key distribution and management is a primary research area in wireless sensor network. The requirement of secure and efficient communication between the nodes in a sensor network has directed the cryptographers to design various light weight key encryption and distribution techniques. In this paper, they have discussed and analyzed the different key distribution protocols and proposed the highly secure polynomial pool based key distribution technique with the help of sparse matrix to reduce the storage and computational complexity.

In [**12**], the authors presented security of wireless sensor networks which is a major issue to protect sensor nodes from the attacker. Key management or key pre-distribution (KPS) is the preliminary step in the security of a sensor network, where secret keys are loaded into the sensor node's memory before placement of the node in target positions. Various schemes for KPS have been proposed for location dependent as well as location independent wireless sensor networks. Costas arrays are\ (n\times n\) matrices grid represented where dots are placed for the l's and leave blanks for the 0's of the matrix for a positive integer n with the property that the vectors connecting two dots of the grid are all distinct as vectors. Costas arrays have a wide range of application including the construction of sonar signal pattern, cryptography, etc. Due to the Costas arrays property of uniqueness among their elements, it can be used in key pre-distribution of wireless sensor networks, especially in a location-dependent grid-based network. In this paper, they proposed a Costas array based key pre-distribution scheme. Simulations are done on different network scenarios and the results were analyzed. Comparative performance analysis of the scheme is shown for different grid sized network considering various order of Costas array.

Problem statement:

The IoT based wireless sensor networks (WSNs) have a set of security requirements that must be achieved to protect the networks against most of the associated attacks. These security requirements are confidentiality, integrity, availability, authentication, and refreshment. To provide these requirements, a key management mechanism suitable forWSN must be implemented.The key management in WSN is a set of key distribution mechanisms; each mechanism is responsible for establishing cryptographic key or key material among all sensors nodes in the network. In WSN, lifetime of network determine by total energy consumed by nodes. Therefore, nodes are grouped into clusters, in which cluster heads (CHs) collect the data. Security is one of the most important issues in WSN. A good key distribution mechanism should have the following features: scalability, efficiency, connectivity, and resilience. The proposed method EECLDSA algorithm applies for security purpose in cluster head.

III. PROPOSED MODEL

The Secure energy efficient protocol (SEEP) involves two algorithms such as M-ZEP algorithm

In which nodes communication is implemented in hybrid mode, where normal nodes communicate with the sink or base station directly, while advanced nodes are clustered and transmit data to BS through cluster heads. and whereas E-SHA256 (Algorithm1) helps in key generation for each cluster to ensure security of the network and for key updating for each transaction of data.

Proposed Threshold of MZ-SEP:

The proposed protocol is a hierarchical clustering heterogeneous routing protocol known as Modified zonal-stable election protocol (MZ-SEP) routing protocol with hierarchical clustering approach for wireless heterogeneous sensor network (MZ-SEP). This algorithm is an extended version of Z-SEP algorithm, that prolongs the lifetime of the sensor network. We improve the threshold value $T(n)$ of the parent protocol to residual energy-aware cluster head selectionalgorithm. Thus, the new modified $T(n)$ equation is derived as:

$$T(n)=\begin{cases}\dfrac{p}{1-p\times\left(r \bmod \left(\dfrac{1}{p}\right)\right)}\times\dfrac{E_{res}}{E_0}\times K_{opt}, for\ n\in G \\ 0, for\ n\notin G\end{cases} \quad \textbf{(1)}$$

where E_{res} is the nodes remaining residual energy, and E_0 is the initial level of the supplied energy. The optimal cluster number K_{opt} can be given as in [13]:

$$K_{opt}=\sqrt{\frac{n}{2\pi}}\sqrt{\frac{E_{fs}}{E_{amp}\times d^4(2m-1)\times E_{Tx,Rx}-mE_{DA}}}\times M \quad \textbf{(2)}$$

where n is the node number, d is the distance to the base station, m is the quantity of cluster heads associates, $E_{Tx,Rx}$ is the tranceiving/receiving energy, EDA is the data aggregation energy, E_0 is the assigned initial power of each sensor node and M denotes the diameter of the deployed network. Nodes in the network spend definite amount of energy after the data transmission, which varies according to the distance d between the transmitting and receiving node.

Protocols usually use a randomly organized allocation of nodes in the deployed area, thus the inefficient use of nodes energy can be observed. In MZ-SEP algorithm, the deployed area is separated into three fields as zone 1, zone 2 and zone 3 to optimize the energy consumption at the nodes. To distribute the network load and due to nodes assigned initial energy value, nodes are categorized into two groups as normal and advanced nodes. Advanced nodes have more energy than normal nodes. Therefore, normal nodes are deployed in zone 1, close to base station, while advanced nodes are deployed distant to base station (BS) in zone 2 and zone 3. We suppose that nodes are not mobile in the field and field dimensional is unknown. Assume m as a fraction of total nodes n with a times more energy. Thus, $(1 - m)n$ of normal nodes. Normal regular nodes transmit records to base station directly, whereas in advanced nodes hierarchical clustering is formed among nodes, where one sensor node is elected as a cluster head and other cluster members join and deliver data to this cluster head, and then this cluster head transmits aggregated data to BS as shown in Figure 2.

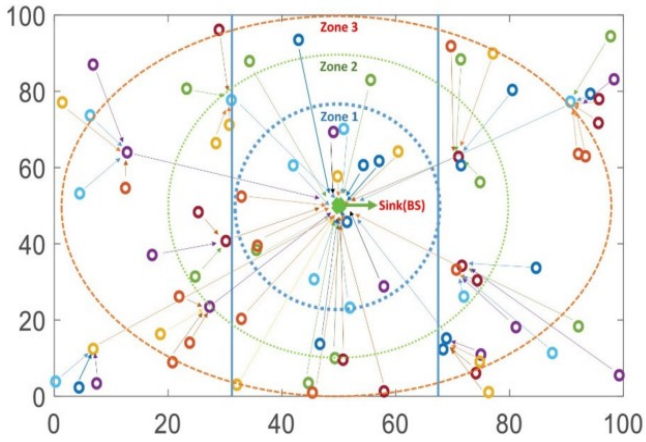

Figure 2: Nodes connection to base station

Key management with MZ-SEP algorithm:

Secure key management is ensured by Modified zonal-stable election protocol (MZ-SEP) algorithm. The security key is generated and maintained between the nodes throughout the lifetime of wireless sensor network. Data authentication, integrity, involves the use of key to transmit data in a secured way. Each CH carries out MZ-SEP algorithm. The CH sends request, which comprises of key and CH node ID. BS validates ID received from all CHs with the database to avoid any fraudulent user. CH always stores a database of its ID and zonal field of every sensor network. A key generated for each transaction and it will get updated each time a data crosses one CH to other and the change in the node status will be updated to BS.

This section presents the proposed secure key management technique and describes its different aspects. Here we present the process when the message is sent through an intermediate node (see Figure 3).

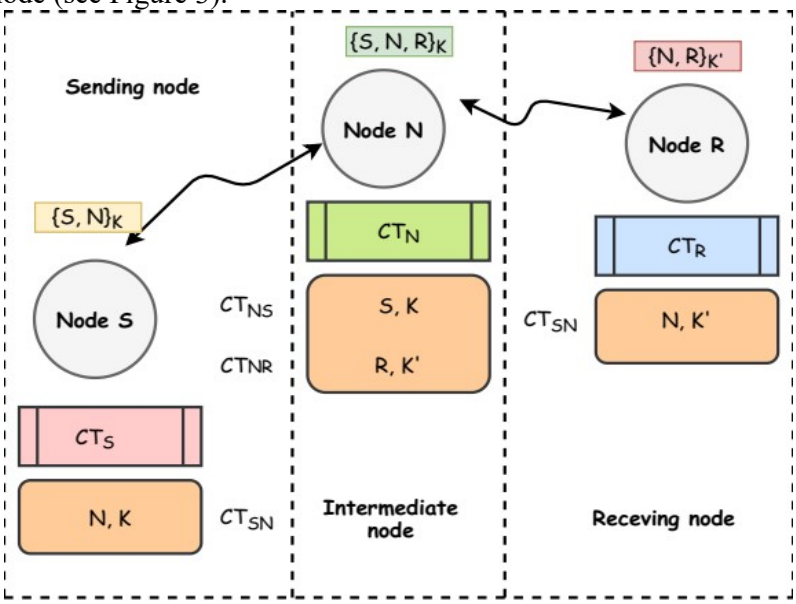

Figure 3: Transfer of messages via an intermediate node N

1. In this scenario node S sends a message m through an intermediate node N to end node R. So for bidirectional connections between S and N, N and S have CT_{SN} and CT_{NS} respectively. Similarly, for bidirectional connections between N and R, R and N provide CT_{NR} and CT_{RN}.
2. S uses the $\{S, N\}_K$ link to send a message to N, and N uses the $\{N, B\}_K$ link to send a message to R.
3. Now, if S wants to send a message to R, it will first pair with S and N with the same mechanism we talked about in the case of a direct link. When the message is received by N, it combines with R to send the message in the same manner to R. Eventually, the $\{S, N, R\}_K$ secure communication network is enabled.

The methodology can be applied to any node series where there exist three or more nodes (S, N_0, N_1, ..., N_r, R). Ultimately, we can claim that the message sent from node S to node R can be safely communicated via an arbitrary list of intermediate nodes N_0, N_1, ..., N_r.

Property: Polynomial based pairwise scheme

Let Zp be a finite field, p prime Let $P(x, y)$ be a symmetric polynomial of degree c with coefficients in Zp

So, $P(x, y) = P(y, x)$

Node N_i receives the polynomial $f_i(x) = P(x, i)$

Node N_j receives the polynomial $f_j(x) = P(x, j)$

Common key between i and j is $P(i, j) = P(j, i)$

Key Generation and Distribution Process:

This section discusses the key generation and distribution processes (see Algorithm 1). According to this algorithm, if any pair $\{S, R\}_K$ does not exist, then it checks whether the connection between S and R exists or not. If a connection exists in CT_S, then sender S does not need to generate K as it can reuse the key K that is stored in its global table. However, if the connection does not exist, it means that it is a new connection and needs to generate a new mutual key K. The sender generates the key K and updates its global table and finally creates a hash key k by applying a key conversion function. The algorithm now checks the reverse connection from R to S. If the connection exists, it means $CT_{RS}$and LT_R need to update with the new key. K and k will be stored in CT_R and LT_Rdirectly if the connection does not exist. In this way, a secure key pairing will be created, and a secure message will be sent from the sender node S to receiver node R by calling the Algorithm 2.

Algorithm 1: *SIoT* (S, R, *msg*)

Input: Any sender node S, and any receiver node R and a message *msg*

Output: The algorithm will try to send a message m from sender S to receiver R. If the operation will be completed successfully it will return true otherwise return false.

/* S,R are defined as disconnected when the algorithm fails. */

if not exists, $\{S, R\}_K$**then**

if not exists, CT_{SR}**then**

/* a new mutual key K will be generated by S */

K = *generateKey*()

/* S add its CT_S and GT_S with K and a random value v */

addCT(R, K);

addGT(K, v);

/* make the hash key k using the conversion function */

$k = f(A, K, v)$;

if CT_{RS} exists **then**

/* *R* receive *k* and *v* from *S* and update the existing entries for CT_{BS} and LT_R if the stored hash match */
updateCT(*S*, *K*);
updateLT(*K*, *k*);
else
/* *R* receive *k* from *S* and add to LT_R and CT_R */
addCT(*S*, *K*);
addLT(*K*, *k*);
/* The secure registration is available so find the *K* in CT_S or CT_R and finally send the message, *msg* */
stat ← *sendMessage*(*S*, *R*, *msg*, *K*);
return *stat*

Key Update Process:
In many cases, with the new value, nodes need to change the local key. A node can trigger an update message when the local key has to be changed so that all connected nodes can modify their tables. The main mechanisms for updating are as follows:
1. Node *A* wants to change all its keys or a specific key. In the first stage, *S* checks for CT_S for all key entries, then submit message change commands to all link nodes with *S* that share the same key.
2. Now say *R* gets a *S* change message. Using the hash key k and the value v that it previously stored in LT_S and GT_R tables, verifies this change message to ensure security. So *R* authenticates *v* with *k*. If no match is found, the key change request will be discarded by *R*. If found, then *R* updates its key entry in all associated tables.

Algorithm 2: sendMessage (*S*, *R*, *msg*, *K*)

Input: Sender node *S*, receiver node *R*, message m and mutual key *K*

Output: True if the message sending operation completed successfully, otherwise return false
/* perform encryption operation */

$msg^e \leftarrow$ extract control part from m and encrypt the control part using *K*; Finally merge the encrypted control part with the message *msg* .

/* Node *S* transmits the encrypted message to node *R*, where *R* receives it and decrypts the control part */

/* If *R* receives the message successfully then return a positive acknoledgment */

ret ← transmit (*S*,msg^e, *R*)

return ret

IV. RESULTS AND DISCUSSION

The simulation plays an important role in demonstrating IoT-based solutions. In our case, we to show the practical usages of the proposed key management technique by integrating physical SO with simulation data. In addition, taken into consideration different evaluation scenarios and analyzed the output results of our protocol comparing with other given protocols in terms of performance behavior. Network model parameters considered for

MATLAB simulations. In our simulator, we used the SHA256 bit hash function to generate the key. A hash key is created whenever a new connection is established. So the performance of our algorithm depends on the hash function creation.

Effect on Different Key Conversion Function:

In our algorithm, every sender node generates a key, K, when it receives a demand for a global key from the next receiver node. The sender node sends the key to the receiver node as well as makes a secret string which is represented in our algorithm as key conversion function ($k = f(v)$, where v is a value that is stored in the global key table of sender node).

The purpose of this one way hash function is when an attacker requests the key, the receiver can use authentication using v and k. As for every key generation time, the system needs to execute this function, so overall performance depends on the execution time of this function. We tested our algorithm based on different key conversion functions, and the result is demonstrated in figure 4. According to this figure 4, we see that SHA512 is more secure, but in our simulator, we used the SHA256 hash function to generate the hash key k because it takes less time than SHA512.

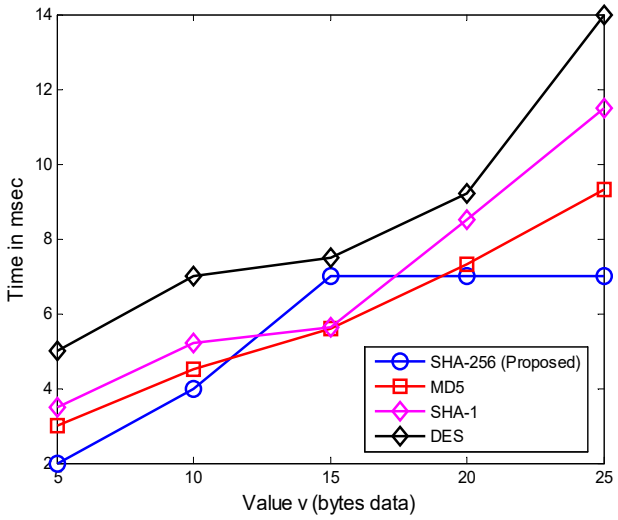

Figure 4: Effect of the different key conversion function.

Key Generation and Distribution Time:

For key generation and distribution, we used Algorithm 1. We set up our model to evaluate the processing time of this algorithm. When we started our experiment, all the three nodes created their private and public key pairs and shared their public keys to other nodes so that any node can use the appropriate public key for message transfer.To be uniquely identified, each node is assigned a unique id. To get the average processing time, we ran the process ten times. Figure 5 indicates the impact of our experiment. We found from this experiment that the algorithm would cost about 17 ms.

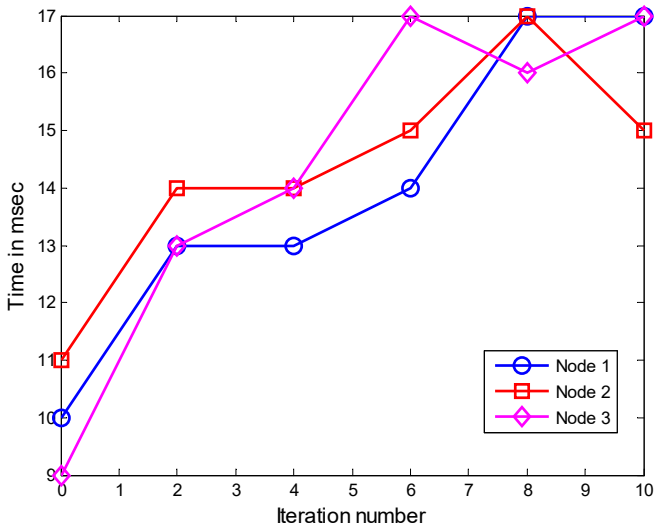

Figure 5: Key generation and distribution time

Analysis of key management with MZ-SEP protocol:

Figure 6 clearly shows that our protocol is enhanced from SEP and LEACH in terms of stability. As LEACH is very sensitive to heterogeneity so nodes die at a faster rate. SEP performs better than LEACH in two level heterogeneity, because SEP has weighted probability for selection of cluster head for both normal nodes and advance nodes. MZ-SEP performs better than LEACH and SEP, because nodes in Zone 0 (normal nodes) communicates directly to base station while nodes in head zone 1 and head zone 2 communicates via cluster head to base station: As in clustering technique, cluster head consumes energy in the form of data aggregation and also by receiving data from nodes in the cluster. So this energy is conserved in normal nodes as they do not have to aggregate data and receive data from other nodes, so energy is not dissipated as that of cluster head, resulting the increase of stability period.

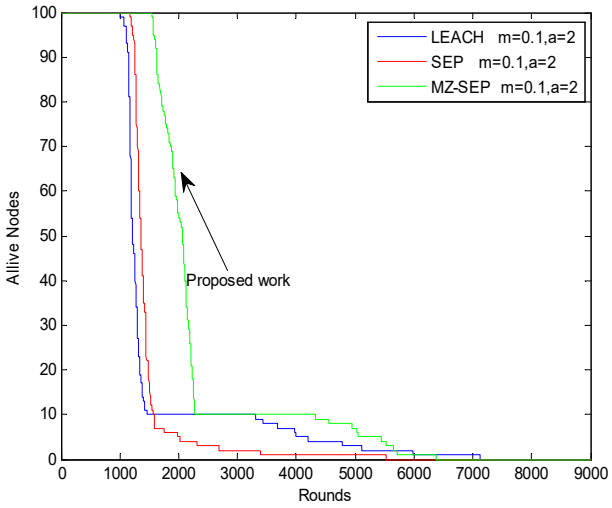

Figure 6: Alive nodes in LEACH, SEP and Proposed MZ-SEP

Generally speaking, the number of dead nodes in the EZ-SEP algorithm is less than in the other three protocols as the network continues to operate. Death of all nodes in parent

protocols is considered at 8000 rounds, while the proposed EZ-SEP protocol shows improved results over 9000 rounds, thus significantly improving the whole network lifetime as shown in Figure 7.

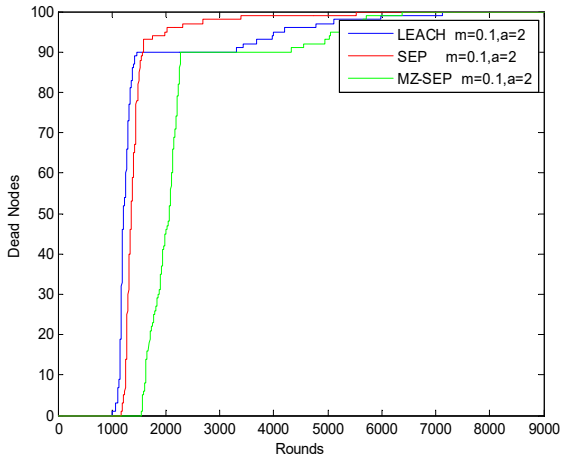

Figure 7: Dead nodes in LEACH, SEP and Proposed MZ-SEP

Network packet delivery ratio to a BS is illustrated in Figure 8 as a throughput. Nodes' data transmission to a base station is the main role of a WSN. More data will be sensed and more environmental information is formed and delivered to the sink, if the network is alive for a long time, thus making the network efficiency more convenient for long time usage. Given below is the graph of all four protocols data packet delivery to BS. It is easily seen in the graph that proposed the MZ-SEP algorithm's performance is much better than that of the other protocols by transmitting more than 2.4×10^5 packets to BS. The other three compared protocols show their incapability to improve upon the proposed new algorithm, as illustrated in Figure 8.

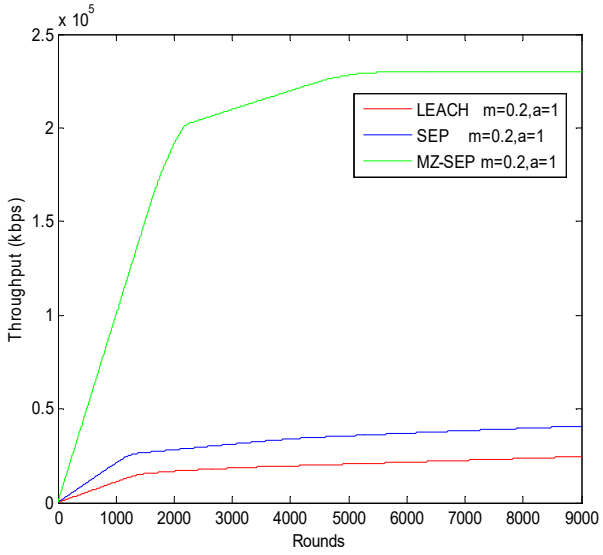

Figure 8: Packets delivery ratio to BS

V. CONCLUSION

IoT is rapidly expanding over the Internet, where a safe communication system is essential. In this paper, we presented an approach defining Smart Objects (SOs) that handles heterogeneous data sources to provide a robust and unified representation of data and to ensure the level of security and reliability associated with each data object. The prototype environment demonstrates the key generation and exchange of IoT data between two physical SOs, while the simulated environment gets data from the implemented prototype within a smart home scenario consisting of multiple nodes. During our systematic and simulation results, we have exposed that our proposed scheme requires a constant number of hash values to calculate a polynomial whereas in existing schemes that number increases rapidly with cluster size. We noticed that the proposed approach is appealing and performs well in terms of key conversion time, algorithm execution time, and bandwidth utilization.In conclusion, three different protocols like SEP and LEACH have been observed and compared to the proposed MZ-SEP algorithm, where MZ-SEP demonstrates better performance reducing the dead nodes by 48%, increasing the packet delivery ratio by 16% and decreasing the average power consumption by 34%.

REFERENCES

[1] Sreeja, B & Loganathan, Jayakumar & Saratha, G. (2018). Wireless sensor network applications: A study. International Journal of Pure and Applied Mathematics. 118. 385-389. 10.12732/ijpam.v118i11.47.

[2] Wosowei, Julius & Deemed, Jain. (2021). Underwater wireless sensor networks: applications and challenges in. International Journal of Current Advanced Research. 10. 23729-23733. 10.24327/ijcar.2021.23733.4705.

[3] Charan, Kamana & Nakkina, Harsha & Chandavarkar, B. (2020). Generation of Symmetric Key Using Randomness of Hash Function. 1-7. 10.1109/ICCCNT49239.2020.9225280.

[4] Madhira, Srinivas & Sammulal, Porika. (2014). Survey on Symmetric and Asymmetric Key Cryptosystems. IOSR Journal of Computer Engineering. 16. 11-18. 10.9790/0661-16431118.

[5] Liu, Wei & Xie, Zhenwei & Liu, Zhengjun & Zhang, Yan & Liu, Shutian. (2015). Multiple-image encryption based on optical asymmetric key cryptosystem. Optics Communications. 335. 205–211. 10.1016/j.optcom.2014.09.046.

[6] Ghasemzadeh, Hamzeh & Payandeh, Ali & Aref, Mohammad. (2017). A Hybrid DOS-Tolerant PKC-Based Key Management System for WSNs.

[7] Ghasemzadeh, Hamzeh & Payandeh, Ali & Aref, Mohammad. (2017). Key management system for WSNs based on hash functions and elliptic curve cryptography.

[8] Amarnath, J. & Shah, Pritam & Chandramouli, H. (2020). Advanced Key Management System (AKMS) for Security in Public Clouds. 10.1007/978-981-15-5788-0_55.

[9] Souza, Rick & Lung, Lau & Custódio, Ricardo. (2013). Multi-Factor Authentication in Key Management Systems. 746-752. 10.1109/TrustCom.2013.90.

[10] Zhang, Y. & Xu, L. & Huang, X. (2012). Polynomial-based key pre-distribution scheme in wireless mesh networks. Journal of Computational Information Systems. 8. 2539-2549.

[11] Choudhary, Vishal & Taruna, Sunil. (2020). The highly secure polynomial pool-based key pre-distribution scheme for wireless sensor network. Journal of Discrete Mathematical Sciences and Cryptography. 23. 95-114. 10.1080/09720529.2020.1721880.

[12] Saikia, Monjul & Hussain, Anwar. (2020). Costas array based key pre-distribution scheme (CAKPS) for WSN and its performance analysis. CSI Transactions on ICT. 8. 10.1007/s40012-020-00300-9.

[13] Hussain, S.; Matin, A.W. Energy efficient hierarchical cluster-based routing for wireless sensor networks. In Technical Report; Jodrey School of Computer Science Acadia University: Wolfville, NS, Canada, 2005; pp. 1–33.