Computer Systems Science & Engineering
DOI:10.32604/csse.2022.021052
Article

Tech Science Press

Cost Effective Decentralized Key Management Framework for IoT

Raja Lavanya*, K. Sundarakantham and S. Mercy Shalinie

Department of Computer Science and Engineering, Thiagarajar College of Engineering, Tamil Nadu, India
*Corresponding Author: Raja Lavanya. Email: rlit@tce.edu
Received: 21 June 2021; Accepted: 04 August 2021

Abstract: Security is a primary concern in communication for reliable transfer of information between the authenticated members, which becomes more complex in a network of Internet of Things (IoT). To provide security for group communication a key management scheme incorporating Bilinear pairing technique with Multicast and Unicast key management protocol (BMU-IOT) for decentralized networks has been proposed. The first part of the proposed work is to divide the network into clusters where sensors are connected to and is administered by cluster head. Each sensor securely shares its secret keys with the cluster head using unicast. Based on these decryption keys, the cluster head generates a common encryption key using bilinear pairing. Any sensor in the subgroup can decrypt the message, which is encrypted by the common encryption key. The remaining part focuses to reduce communication, computation and storage costs of the proposed framework and the resilience against various attacks. The implementation is carried out and results are compared with the existing schemes that have given considerably better results. Thus, the lightweight devices of IoT can provide efficiency and security by reducing their overhead in terms of complexity.

Keywords: Bilinear pairing; common encryption key; key distribution

1 Introduction

Internet of Things (IoT) is developing quickly in the last decades and keeps on advancing in terms of dimension and complexity. IoT prevails in various application domains such as agriculture, space, healthcare, manufacturing, construction, water and mining. It is a powerful element of next generation wireless sensor network which comprises of low power devices. It enables communication of confidential data among these low power devices which are interconnected in anonymous and untrusted environment [1]. The communication of data among the untrusted devices leads to various security breaches. The security threats and vulnerabilities of IoT devices, which are featured with resource constraints and low transmission rates, are increasing day by day. This brings the need for security solutions to ensure confidentiality and authentication. Since the computational capacity of IoT devices are very low, cryptographic operations to ensure confidentiality and authentication faces many challenges. Hence we require an energy efficient smart key management [2] with low communication and computation cost to ensures confidentiality and authentication. It is one of the techniques to certify security against various

types of attacks such as inside, outside attacks. Such techniques must also ensure requirements of key management such as forward secrecy, backward secrecy, key independency and group secrecy [3]. It includes various modules such as key generation, key distribution and rekeying [4,5].

Key management scheme has to satisfy various security goals such as confidentiality and authentication. In an IoT application, many sensor nodes are linked together and information needs to be transferred. A group key is created among the sensor nodes which can be used for communication between the authenticated group members. The group key has to be refreshed every time the nodes join and leave the group. This process requires high computation, communication and storage cost. Many key management schemes exist which have provided solutions to reduce the various types of cost mentioned above. Still efficiency cannot be achieved while reducing these cost in the lightweight devices used in Internet of Things environment. The research issue that remains in the dynamic resource constrained environment is the number of computations has to be reduced even if the number of users increases.

There are many existing methods of key management provided by the researchers. Here we propose a decentralized key management for Internet of Things application for improving security. Bilinear pairing algorithm is used for key generation and the key distribution phase involves the multicast–unicast approach for sharing their private keys and group keys.

The contributions of the proposed work are explained as follows:

- A cost effective decentralized key management framework using Bilinear pairing and Multicast Unicast (BMU-IOT) key management methods is proposed.
- Our approach is tested in a simulation environment of Contiki OS with Cooja simulator.
- The results of the proposed work are compared with existing schemes that shows improved efficiency.

Structure of the Paper

The structure of the paper is organized as follows: In part 2 we briefly discuss about the survey of the existing methods of key management. Part 3 deals with the system model and problem statement. Part 4 deals with the possible solution to overcome that. It includes key generation and distribution. Part 5 shows the analysis of BMU-IOT with the existing approaches. The security analysis is discussed in Section 6. Part 7 concludes the proposed work.

2 Related Works

Over the past decade dynamic nature of group communication has been susceptible to various key security issues such as confidentiality, integrity and authentication. Key management is a technique which ensures all the security issues of group communication. A detailed survey of various key management schemes is discussed here. Key management schemes are divided into centralized, distributed and decentralized schemes [6–8]. All these key management techniques are used for group communication.

In centralized key management scheme, a centralized authority is responsible for supervising the entire network. This scheme also has a bottleneck in security attributes, and network resources [9]. Moreover, the cost of rekeying is higher due to the rekeying of intermediate keys. In batch based group key management a centralized approach has been proposed for key management. Centralized server KDC is responsible for key generation which leads to high computation and communication cost [10]. Logical Key Hierarchy (LKH) is a centralized framework of key management that was proposed by Wong et al. All the keys are mapped to a hierarchical tree where each level corresponds to different nodes of the tree. The intermediate nodes possess Key Encryption Keys (KEK) and the root possesses Traffic Encryption Key (TEK). The drawback of LKH is that during rekeying all the members are involved on key generation, which leads to high communication cost. OFT (One –way Function Tree) is based on symmetric keys. Many intermediate keys are rekeyed

which results in high computation cost. Kung et al. proposed a centralized key management system which uses LKH protocol. Since the key management is based on symmetric keys, all the users of the group are involved in updating the keys during rekeying process. This lead to increase in computation cost during rekeying [11–14].

In decentralized key management scheme, the entire group is split into various subgroups, which are administered by subgroup controllers. This key management scheme eliminates the bottleneck problem and single point vulnerability problem. It significantly reduces the rekeying cost when compared to centralized scheme [15]. Abdmeziem et al. [16] proposed a decentralized batch-based group key management which includes various subgroups. Since all the subgroups are managed by different servers the computation cost and storage cost increases. Tsai et al. [17] proposed a lightweight key management protocol which involves the Kronecker product. This method does not ensue forward and backward secrecy. Multicast–Unicast Key management and Distribution (MUKD) framework is another decentralized key management approach. In this framework also, a group is split into various subgroups and each subgroup is controlled by Subgroup Manager (SM). Dividing the network into subgroups, addresses single point failure. The key management process, which includes key generation, key distribution and key refreshment are managed by the Subgroup Manager. MUKD distributes the newly generated keys. The drawback of this approach is that the communication between the subgroup requires many encryption and decryption for key update [9]. In distributed key management scheme, every member of the group must share their contribution for generation of group key. Since all the members of the group contribute for the group key generation, the cost of rekeying is high. Abdmeziem et al. proposed a distributed key management scheme which had high communication cost due to rekeying [18].

From the above described key management techniques it is inferred that the majority of existing key management scheme has various drawbacks in terms of communication, computation and storage cost. They are also prone to various attacks. Since IoT devices are resource constrained, the computation and storage cost has to be reduced. Hence we propose a cost effective decentralized key management for Internet of Things application for improving security. The following Tab. 1 summarizes the essential features of the existing key management techniques: (i) Forward secrecy, (ii) Backward secrecy. These features ensure the key updation during join and leave operations. (iii) Key independence ensures that keys are independent of one another (iv) Single point failure and (v) Scalability

Table 1: Comparison of existing methods

	Centralized	Decentralized	Distributed
Forward secrecy	No	Yes	Yes
Backward secrecy	Yes	Yes	Yes
Key independence	Yes	Yes	No
Single point of failure	Yes	No	No
Scalability	No	Limited	Limited

3 System Model

This subdivision sketches the network model and then discusses the problem statement of the proposed work.

3.1 Network Model

The basic architecture of IoT defines the functional framework of various components. It has three layers namely: Perception layer, Network layer and Application layer [19]. These layers depict the functional

organization and configuration of the components involved. The security issues and the characteristics of the layers are demarcated as follows and portrayed in Fig. 1.

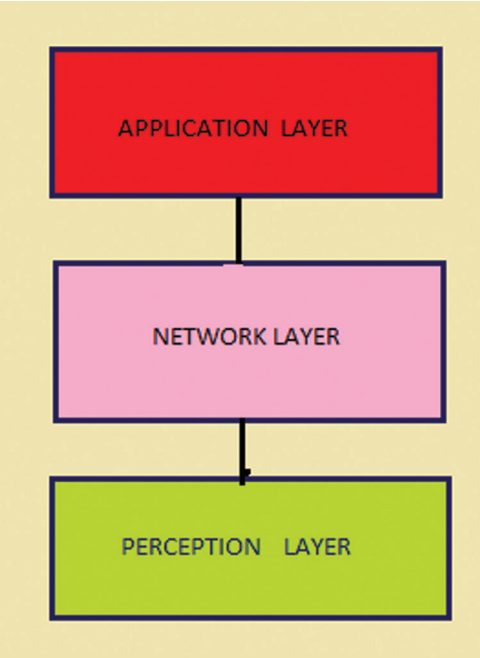

Figure 1: IoT architecture layers

3.1.1 Perception Layer

This is the physical layer, which comprises of sensors and actuators. This layer detects, gathers, and processes information received from the environment and then communicates it to the network layer. Security issues in perception layer includes node authentication, node capture. Every sensor node has to be authenticated to stay away from counterfeit node and illegal access. Confidentiality of data has also to be maintained while data is being transmitted across different nodes. The perception layer is also prone to various malicious attacks.

3.1.2 Network Layer

Network layer is liable for communicating the data gathered by the perceptual layer to any other IoT device or to a cloud server. The data is transmitted securely using secure communication protocols, mobile networks and various other technologies like WiFi, Zigbee. The various security issues that still persist in this layer include routing problem, congestion problem and spoofing.

3.1.3 Application Layer

This layer serves the users request as per their needs. At times, it serves as a part of cloud servers, which allows large IoT applications to integrate and build analytic solutions. Authentication and access control mechanisms are vital to ensure application layer security. The Fig. 1 shows the various architectural layers of IoT.

3.2 Problem Statement

The security of the IoT architecture is essential to the development of various applications. Sensing and communication of data between the architectural layers faces security issues such as node authentication and

confidentiality of data. To overcome these security issues we require a novel cost effective decentralized key management architectural framework for improving the efficiency and security in the IoT network. This can be achieved using lightweight multicast and unicast key management schemes. The combination of these two communication methods allows the usage of one unique key for encryption and multiple keys for decryption. This approach ensures security against various attacks and is used to minimize the communication, computation and storage costs during group membership changes of the nodes in the network.

4 Proposed Decentralized Framework

The proposed decentralized framework for secure communication is described in terms of elements of the framework and the way of communication between the elements of the framework.

4.1 Elements of the Framework

A similar network model of three layered architectures is followed. The proposed decentralized network model comprises of the following elements as described in the Fig. 2. The decentralized architecture is simulated as a tree hierarchy with different levels. The resource constrained sensor nodes (SN) are present at the lower level of the tree. The sensor nodes of certain functionality are grouped together as separate sub groups. Each subgroup of sensors needs to have a Cluster head (CH). The sensor node with high capability is elected as the Cluster head (CH). The most powerful Gateway (GW) serves as a communication bridge between sensors and other devices, which are in the next higher level of the tree. The cluster head of the subgroup is connected to the Gateway. The task of the Gateway is to transmit the data securely to the Cloud Server, which is considered the root of the tree.

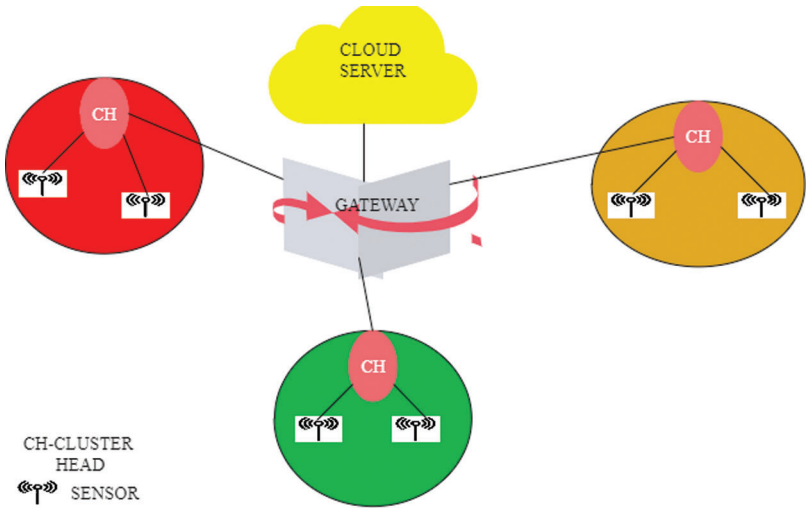

Figure 2: Architecture diagram

4.2 Communication Between the Elements

As the IoT devices proliferate, the communication between the sensor node and cluster head, cluster head and gateway are done in wireless medium. Security aspects such as confidentiality and authentication has to be ensured in the communication channel. In the above described framework both unicast and multicast communication takes place. In unicast method, the sensors and the cluster head, the cluster head and gateway, the gateway and cloud server communicates with each other individually in one to one fashion. The cluster head multicast a message to all the sensors of its cluster in one to many fashion.

4.3 BMU-IOT Key Management Method

In this section we discuss about the precise concepts of the proposed key management framework. Consider a dynamic scenario in which the data has to be communicated securely from the sensor nodes to the cloud and confidentiality of data has to be maintained. BMU-IOT is one such method which generates lightweight group key to encrypt the data and thus ensure confidentiality. BMU-IOT uses the following approach for subgroup formation with the Cluster head. Each sensor generates their Secret Keys (SCK) and shares with the Cluster head in a secure unicast method. The Cluster head will formulate the common encryption key using the shared secret key, which is used for group communication. The sharing of common encryption key is not required for this communication. Any sensor having a place with that cluster who shares their SCK can decode the message from Cluster head. If a new sensor joins the cluster or if an existing sensor leaves the cluster, a new encryption key is formulated using the secret keys of existing sensors. The formulated common encryption key can be used as a group key, which is distributed through polynomial expression in a multicast manner. The sensors can retrieve the group key from the polynomial expression by substituting their secret keys. The distinct phases involved in the BMU-IOT key management process is sketched in detail as follows:

- Registration of devices
- Cluster formation among the sensors
- Key generation using bilinear pairing
- Key distribution

4.3.1 Registration of Devices

The proposed decentralized framework encompasses various resource constrained devices. At the initial stage each device has to register with the network. The resource constrained sensor nodes and gateway register themselves with the unique product id to the cloud server before linking the wireless communication channel. Hence only authenticated devices are permitted to join the unambiguous network. The cloud servers ensure the mutual authentication of devices and proceeds to next phase of cluster formation among the sensors.

4.3.2 Cluster Formation Among the Sensors

The resource constrained sensors that have insufficient processing power and limited battery capacity are responsible for gathering real-time data from various physiology metrics. Authorized sensors that collect relevant data and perform similar functions constitute a subgroup. It is essential to have a cluster head for each subgroup to manage the sensor data. The sensor node that is largest weighed and has greater computation ability is elected as the Cluster Head (CH). The CH of each subgroup is connected to the gateway and the gateway is connected to the Cloud Server, which is considered as the root of the tree.

4.3.3 Key Generation Using Bilinear Pairing

Sensor nodes belonging to a subgroup share their unique identifiers with the cluster head. The secret identifier used to authenticate the sensor is sent securely on the unicast channel. All these values are stored securely in the table. The cluster head uses bilinear pairing for the generation of common encryption key. The following parameters is used for bilinear pairing. Every sensor node has a secret key SCK_i where $i = 1, 2, 3, 4, 5, \ldots n$ is shared with the cluster head. The cluster head computes the group key, which is the common encryption key (CEK). Any sensor, can decrypt the message by using its SCK, which is being encrypted by CEK. The algorithm for key generation is explained as follows:

Algorithm 1: Key Generation

1. Let G_1 and G_2 be abelian groups.
2. Let n be a prime number such that [n]P for all P in G_1 and G_2.
3. Let G_p be a cyclic group of order n.
4. The CH and sensors choose G_p as a fixed group and g is the generator of that group.
5. Let e: G_1 X G2 → G_p be bilinear and g is the generator of that group.
6. A sensor from G1 unicasts 'ida', G2 unicasts 'idb' to CH for generation of secret key SCK.
7. The CH collects SCK_i from all the sensors and does the following work:
   The CH selects a random secret value id from {1, 2, …. n} and computes e (g^a, g^b) = e $(g, g)^{ab}$
8. CH issues {CEKi} publicly.

4.3.4 Key Distribution

The bilinear pairing algorithm is used to generate a common encryption key using the sensor ID. Each sensor belonging to this particular subgroup can use this encryption key to perform decryption.

Key Distribution Among Sensors

When the sensor needs to communicate among them, encryption key or group key is essential. So cluster head multicast the common encryption key to the sensors. Message can be encrypted and decrypted using the group key. The cluster head generates a new polynomial to distribute the group key that has the secret encryption key in one of its values and is communicated using multicast channel. Since the secret encryption key is inside the polynomial, it need not be encrypted. The sensors can now derive their group key by using their Secret Key by solving the polynomial equation.

Polynomial Generation:

CH generates the following polynomial using the SCK_i

$$F(u) = (u - SCK_1)(u - SCK_2)(u - SCK_3)(u - SCK_4) \ldots\ldots\ldots (u - SCK_n) + GK_k \quad (1)$$

The given polynomial can be expanded as follows:

$$F(u) = u^n - u^{n-1} + u^{n-2} \ldots\ldots\ldots \pm z \quad (2)$$

Group Key Distribution:

The sensors substitute the secret key value in the expanded polynomial to retrieve the group key where u = SCK. For example, think about the cluster with 4 sensor nodes whose group key is 45. To distribute the group key CH generates the subsequent hidden polynomial, which is as follows:

Consider

$$u = e^x \quad (3)$$

Since there are 4 sensor nodes the polynomial is expanded as

$$F(u) = (u - 1)(u - 2)(u - 3)(u - 4) + 45$$

$$F(u) = u^4 - 10u^3 + 35u^2 - 50u + 69 \quad (4)$$

Then the expanded polynomial equation is embedded in a hidden polynomial H(x) equation before multicasting to the sensors. Therefore, the hidden polynomial equation for Eq. (4) is

$$H(x) = e^{4x} - 10e^{3x} + 35e^{2x} - 50e + 69 \quad (5)$$

Then hidden polynomial H(x) is multicast to the sensors. The sensors have to derive the original value from the hidden polynomial and then substitute their secret value to derive the group key.

Group Key Derivation:

The group key is derived by the sensors using the following procedure:

(1) Derive the original polynomial equation from the hidden polynomial.
(2) Thus the original polynomial equation for H(x) is

$$F(u) = u^4 - 10u^3 + 35u^2 - 50u + 69 \quad (6)$$

(3) Substitute the value of the secret key for every sensor.

If the sensor id is 1, then substitute the value as x = 1 in Eq. (4) to derive the group key. Thus, the equation after substituting the value is as follows:

$$F(u) = u^4 - 10u^3 + 35u^2 - 50u + 69 \quad (7)$$

Therefore

$$F(u) = 45 \quad (8)$$

Hence, the group key value $F(u) = 45$ is derived from the polynomial. The entire process of key generation and distribution are illustrated in the Tab. 2.

Table 2: Key generation and distribution process

Polynomial génération	Group key distribution	Group key derivation
1. Consider the no. of sensors as 4 and 45 is the group key embedded in the polynomial. Thus $F(u) = (u-1)(u-2)(u-3)(u-4) + 45$ *2. The polynomial equation is* $F(u) = u^4 - 10u^3 + 35u^2 - 50u + 69$	*3. The generated polynomial equation is changed as hidden polynomial using the formula* $u = e^x$ *4. The group key is distributed using hidden polynomial H(x)* $H(x) = e^{4x} - 10e^{3x} + 35e^{2x} - 50e + 69$	*5. The original equation is derived from the hidden polynomial H(x) and then substitute x = 1. We get the original equation which is* $F(u) = u^4 - 10u^3 + 35u^2 - 50u + 69$ Sub x = 1 $F(u) = 45$

Key Distribution Among Clusters

The communication among several clusters connected to the gateway also requires secret key for information sharing. All the cluster heads will register themselves with the gateway and the same technique is followed for key generation and distribution.

Algorithm 2: Algorithm for Key Distribution

Input: (members SN_i, Cluster Head CH, Secret Keys SCK_i, Encryption key CEK_K)

1. **For key distribution between sensors**

1.1 Sensor SN_i sends secret key SCK_i to CH.

1.2 CH records SCK_i in its routing table and generates polynomial F(y)

1.3 Equation of $F(y) = e^{\ln((y-1)(y-2)(y-3)(y-4)+GK_k)}$ is used for sensor to sensor communication

1.4 CH then multicast polynomial F(x) to sensors SN_i

1.5 Sensors N_i receives the polynomial F(y) and substitute $y = SCK_i$ on F(x) and derives the CEK_K.

2. **For key distribution between CH and Gateway ((Cluster Head CH_i, Gateway GW, Encryption Key CEK_k)**

2.1 Cluster head shares it group key to gateway.

2.2 GW records CEK_k in its routing table

2.3 GW generates one encryption and multi-decryption set.

3. End

4.3.5 Rekeying

The rekeying process of BMU-IoT depicts the generation of new group key. The key is refreshed and a new group key is formed while a new sensor node arrives or departs from the cluster. Rekeying occurs in the different levels of hierarchy and not all the members are involved for generation of new key.

Sensor Join and Leave:

If a new sensor node request to join a new cluster, it has to perform the following steps:

- The new sensor node initiates a new request to the cluster head.
- The CH_1 will authenticate and forward the request to the gateway.
- The gateway will confirm if it is a new request or a duplicate request by verifying the values of the existing sensors.
- If it is a new request, then the CH will generate a new group key using the id of the new sensor. If it is a duplicate request it is rejected.

Similarly, when a sensor node exits from the group a new secret encryption key is generated by using the shares (SCK) of the existing member nodes.

Algorithm 3: Algorithm for Rekeying

1. **If a new sensor SN_i sends, join request to CH_i**

1.1 CH_i then sends SN_i request to root K_r, which verifies for authentication

2. If SN_i is valid

2.1 Root will accept the request and forwards to CH.

2.2 Then CH_i forwards to SN_i which sends SCK_i using unicast to CH_i and follows same procedure of new group key formation.

2.3 Else it rejects the request

3. **If an existing sensor SN_i sends leave request to CH_i**

3.1 CH_i then sends the request to K_r for authentication and if it is valid

3.2 K_r sends acceptance to CH_i to remove the SCK from its table.

3.3 CH_i generates new encryption key CEK_k" which can be decrypted by the existing members use SCK_i as decryption keys.

3.4 Else it rejects the request

5 Computational Performance Evaluation

The performance of the proposed protocol is estimated with security metrics in various hierarchy levels and compared with the existing protocols. The proposed protocol is simulated using Cooja simulator of Contiki OS which is an open source operating system. Contiki OS is used for memory constrained networking system and low power IoT devices. New motes are created and grouped together as clusters. We deploy 7 clusters of varying sizes for implementing the key management process and various metrics are measured. The metrics used for evaluation are communication cost, computation cost and storage costs of key distribution. We compare and analyze the performance of LKH, OFT, MUKD with our proposed BMU-IoT schemes. The number of sensors (S) at the middle level of the tree or at the intermediate level shows impact on the tree height, which is considered to be H. (n) is the number of users and (d) is the intermediate children.

5.1 Communication Cost

The cost of communication is measured in terms of the numbers of keys for transferring messages during the node join and node leave operations. The messages transferred for unicast and multicast communication are analyzed separately for calculating the cost. The cost of communication for LKH is higher when compared with other schemes since the number of keys and messages transferred rely on the tree height. In OFT the height of the tree is used for the performance analysis where, h is equivalent to $1 + \log d\ n$ since it has binary tree structure. The cost of OFT is also higher since communication cost rely on the height of the tree. The communication cost in the proposed BMU-IoT scheme is determined by the number of messages and keys transferred between the sensors and the cluster head. The cost remains as 1 since the level of the path connecting the sensors to the cluster head is always 1. It provides enhancement in performance when compared with the existing schemes. The following Tab. 3 and Fig. 3 illustrates the comparison of communication costs of various protocols such as LKH, OFT, MUKD and BMU-IOT.

Table 3: Comparison of communication cost

Protocol	Number of message transferred/keys exchanged		
	Join operation		Leave operation
	Unicast	Multicast	Multicast
LKH	$d(\log_d(n) + 1)$	$d(\log_d(n) + 1)$	$d(\log_d(n) + 1)$
OFT	$\log_d(n) + 1$	$\log_d(n) + 1$	$\log_d(n) + 1$
MUKD	1	1	1
BMU-IoT	1	1	1

5.2 Storage Cost

This type of cost determines the total number of keys stored by cluster head and sensors. In LKH and OFT, storage cost is calculated based upon the intermediate children (d) and the users. Hence, the cost is higher based on the different levels of the tree. In the proposed scheme, the storage cost depends on the secret keys stored in the sensors and the encryption key and secret keys stored in the cluster head. Since there are no decryption keys the storage cost of the proposed scheme is less when compared to the other schemes. The following Tab. 4 and Fig. 4 illustrates the comparison of storage costs of various protocols such as LKH, OFT, MUKD and BMU-IOT.

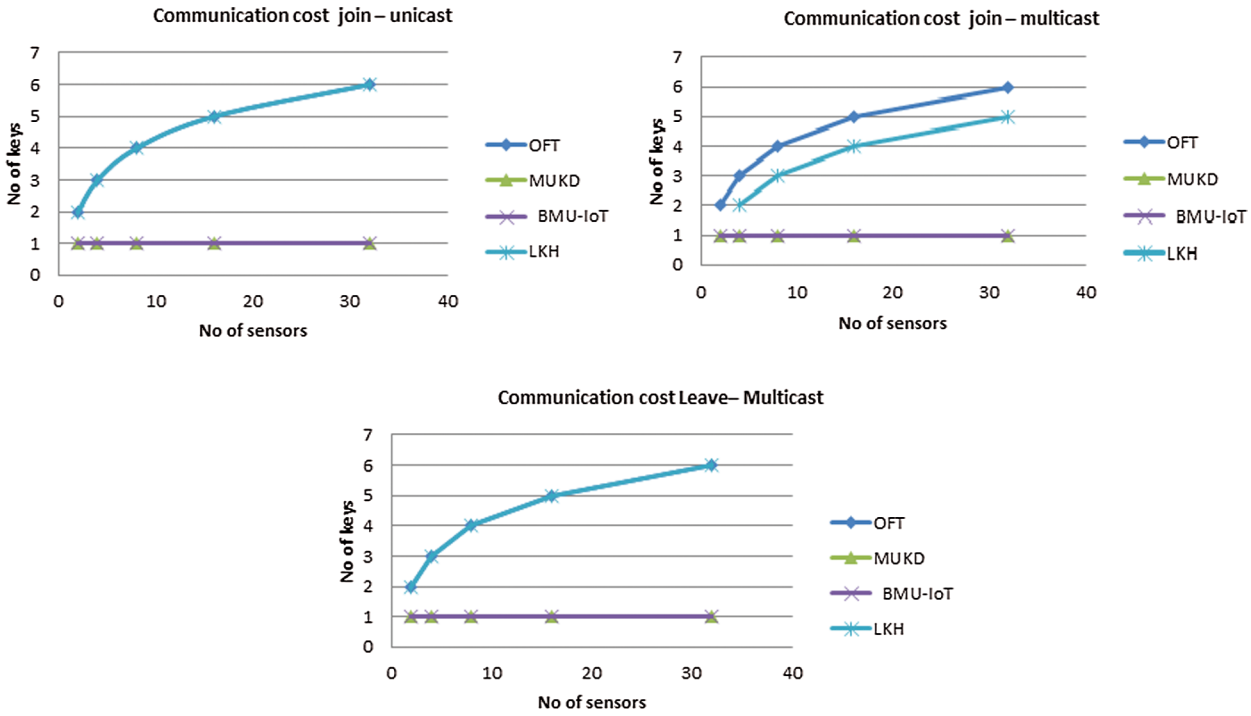

Figure 3: Comparison of communication cost

Table 4: Comparison of storage cost

Protocol	Number of message stored/keys stored	
	Cluster head	Sensors
LKH	$(dn-1)/(d-1)$	$(\log_d(n)+1)$
OFT	$2n-1$	$\log_d(n)+1$
MUKD	k	2
BMU-IoT	k	1

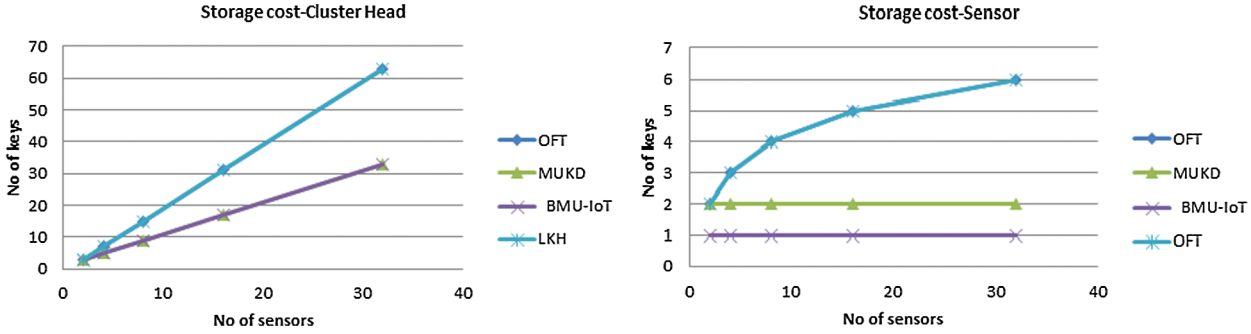

Figure 4: Comparison of storage cost

5.3 Computation Cost

This cost is computed by considering calculation of the encryption and decryption of group key. The computation cost of LKH and OFT is high since all the updates are being performed by the server. The proposed scheme has low computation cost as the server computes the common encryption key using Bilinear pairing and the encryption key is kept at Cluster head itself. The key distribution is not needed for the group communication through CH. For sensor-to-sensor communication, the common encryption key is transmitted to the members using polynomials. The encryption key can be derived from the polynomial by solving it. The generation of polynomial to distribute a group key is more efficient as it reduces the computation in terms of encryption and decryption. The following Tab. 5 and Fig. 5 illustrates the comparison of computation costs of various protocols such as LKH, OFT, MUKD and BMU-IOT.

Table 5: Comparison of computation cost

Protocol	Number of encryption keys at CH		Number of decryption keys at sensors		
	Join	Leave	Join		Leave
			Previous sensor	New sensor	New sensor
LKH	$d(\log_d(n)+1)$	$\log_d(n)$	$d(\log_d(n)+1)$	$d(\log_d(n)+1)$	$d(\log_d(n)+1)$
OFT	$2(\log_d(n)+1)$	$\log_d(n)+1$	$\log_d(n)+1$	$\log_d(n)+1$	$\log_d(n)+1$
MUKD	1	1	1	1	1
BMU-IoT	0	0	0	0	0

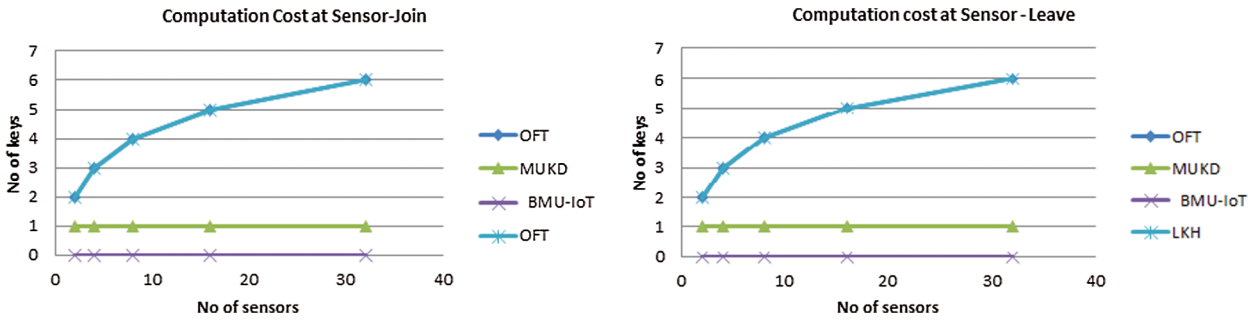

Figure 5: Comparison of computation cost–sensor

6. Security Analysis

Security analysis of BMU-IOT is analyzed in the view of various security requirements of the proposed scheme and its vulnerability against various attacks. The proposed scheme BMU-IOT ensures the security requirements namely: confidentiality, authentication, forward secrecy, backward secrecy, group key secrecy. In BMU-IOT, the CH keeps the common group key. If a sensor wants to decrypt the group key, they can use their own ids, which are unicast to CH. Thus, it ensures confidentiality by not revealing the group key to the attackers. Only the sensors with their own unique ids can decrypt the group encryption key. Any other sensor, which has not shared its id, cannot decrypt the key. Thus, it assures authentication between the sensors.

6.1 Forward Secrecy

This type of secrecy ensures that the keys are not compromised when a sensor leaves the cluster. When the sensor node leaves the cluster, the CH will remove its secret key and it computes a new group key using the existing sensors in the cluster. Hence, the sensor, which leaves the group, will no longer be capable of participating in the future communication. Thus, forward secrecy is ensured [14].

6.2 Backward Secrecy

If a new sensor joins the cluster, the CH computes a new common encryption key or group key with the existing set of sensors. Hence, the communication using the old group key is not possible. Thus, backward secrecy is ensured.

6.3 Group key Secrecy

In the proposed scheme, Bi Linear Pairing is used for the generation of common encryption key. The common encryption key is formulated using the shares of secret key value of all the sensors. The intruders will not be able to compute the group key even if they get the secret key value of any sensor. Thus, group key secrecy is ensured.

6.4 Attacks

The security of the proposed scheme is measured with its vulnerability against several attacks namely internal attack, external attack.

An internal attack is a malicious attack, which is executed on any network by intruders with authorized access. Any sensor in the group who attempts to catch the individual contact of another sensor with CH initiates an internal attack. Since the secret key of the sensor, which is unicasted to the particular sensor, encrypts the data, the intruder will not be able to interpret that communication. An external attack refers to the threat where someone from outside a network tries to exploit device vulnerabilities. Any new sensor or a sensor that has left the group can be considered as an attacker. New sensors who wish to join the group need to be authenticated by the CH so they will not be able to interrupt the communication. Any sensor, which is no longer a member of the group, cannot perform external attack due to forward secrecy.

7 Conclusion

This paper proposes a cost effective decentralized key management framework for secure group communication. The group is divided into various clusters in order to overcome the problems of single point failure. Each cluster has a cluster head who takes the sole responsibility for communicating the group key. In the proposed BMU-IOT, the sensors unicast the Secret Keys and using Bilinear Pairing algorithm, the CH generates the common encryption key. The members can use these secret keys as decryption key. The key distribution is not needed for the group communication through CH. Hence, the communication cost is very low. Since the encryption key is not transmitted to the user, neither encryption nor decryption is involved. Thus, the computation cost of the proposed scheme is proved to be low. The storage cost of the proposed scheme is also low since the number of keys in the cluster head and the sensors is also reduced. Thus, the proposed BMU-IOT scheme has low communication, storage and computation costs. Thus, we conclude that the proposed decentralized key management framework has analyzed the required security considerations of group key management and provides resistance against different kinds of attacks that may occur in group communication.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] T. Gebremichael, U. Jennehag and M. Gidlund, "Lightweight IoT group key establishment scheme using one-way accumulator," in *Proc. Int. Symposium on Networks, Computers and Communications (ISNCC)*, Rome, pp. 265–271, 2018.

[2] M. Verma and D. Huang, "SeGCom: secure group communication in VANETs," in *Proc. IEEE Intl. Conf. on Consumer Comm. and Networking (CCNC)*, Las Vegas, NV, USA, pp. 1–5, 2009.

[3] R. Velumadhava Rao, K. Selvamani and R. Elakkiya, "A secure key transfer protocol for group communication," *Advanced Computing: An International Journal (ACIJ)*, vol. 3, no. 6, pp. 83–90, 2012.

[4] W. Aye and M. U. Siddiqi, "Key management for secure multicast over IPv6 wireless networks," *EURASIP Journal of Wireless. Communication. Networks*, Article ID 61769, vol. 2006, pp. 1–12, 2006.

[5] J. Zhou, L. Sun and X. Zhou, "High performance group merging/splitting scheme for group key management," *Wireless Personal Communication*, vol. 75, pp. 1529–1545, 2013.

[6] G. Vinoth Chakkaravarthy and P. Ambiga, "Dynamic key management schemes: A survey," *Advances in Natural and Applied Sciences*, vol. 8, no. 17, pp. 1–8, 2012.

[7] S. Rafaelli and D. Hutchison, "A survey of key management for secure group communication," *ACM Computing. Surveys*, vol. 35, pp. 309–329, 2003.

[8] X. He, M. Niedermeier and H. De Meer, "Dynamic key management in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 36, pp. 611–622, 2013.

[9] P. Vijayakumar, S. Bose and A. Kannan, "Centralized key distribution protocol using the greatest common divisor method," *Computers and Mathematics with Applications*, vol. 65, pp. 1360–1368, 2013.

[10] L. Veltri, S. Cirani and G. Ferrari, "Batch-based group key management with shared key derivation in the internet of things," in *Proc. 9th Int. Wireless Communications and Mobile Computing Conf. (IWCMC)*, Italy, pp. 1688–1693, 2013.

[11] A. Mehdizadeh, R. S. A. R. Abdullah and F. H. B. M. Ali, "Reliable key management and data delivery method in multicast over wireless IPv6 networks," *Wireless Personal Communication*, vol. 73, pp. 967–991, 2013.

[12] B. Raju and C. Meenu, "A survey on efficient group key management schemes in wireless networks," *Indian Journal of Science and Technology*, vol. 9, no. 14, pp. 1–16, 2016.

[13] J. Bibo and H. Xiulin Hu, "A survey of group key management," in *Proc. Int. Conf. on Computer Science and Software Engineering*, Wuhan, China, pp. 994–1002, 2008.

[14] Y. H. Kung and H. C. Hsiao, "GROUPIT: Lightweight group key management for dynamic IoT environments," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5155–5165, 2018.

[15] M. Dammak, O. R. M. Boudia, M. A. Messous, S. M. Senouci and C. Gransart, "Decentralized lightweight group key management for dynamic access control in IoT environments," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1742–1757, 2020.

[16] M. R. Abdmeziem, D. Tandjaoui and I. Romdhani, "A decentralized batch-based group key management protocol for mobile internet of things (DBGK)," in *Proc. IEEE Int. Conf. on Computer and Information Technology, Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Liverpool, UK, pp. 1109–1117, 2015.

[17] I. -C. Tsai, C. -M. Yu, H. Yokota and S. -Y. Kuo, "Key management in internet of things via kronecker product," in *Proc. IEEE 22nd Pacific Rim Int. Symposium on Dependable Computing (PRDC)*, Christchurch, New Zealand, pp. 118–124, 2017.

[18] M. R. Abdmeziem and F. Charoy, "Fault-Tolerant and scalable key management protocol for IoT-based collaborative groups," *In Security and Privacy in Communication Networks. SecureComm 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Edited by Lin, X., Ghorbani, A., Ren, K., Zhu, S., Zhang, A., Springer, Cham, Switzerland, pp. 320–338, 2017.

[19] I. Yaqoob, E. Ahmed, I. Abaker and T. Hashem, "Iot architecture internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 10–16, 2017.