Secure Pairwise Key Sharing using Geometric Group Key Sharing Method (*Full Paper*)

Shogo Ochiai
Department of Electrical Engineering
Tokyo University of Science
Tokyo, Japan
ochiai_shogo@sec.ee.kagu.tus.ac.jp

Keiichi Iwamura
Department of Electrical Engineering
Tokyo University of Science
Tokyo, Japan
iwamura@ee.kagu.tus.ac.jp

Ahmad Akmal Aminuddin Mohd Kamal
Department of Electrical Engineering
Tokyo University of Science
Tokyo, Japan
ahmad@sec.ee.kagu.tus.ac.jp

***Abstract*— In recent years, the concept of Internet of Things (IoT) network used to enable everyday objects and electronic devices to communicate with each other has been extensively discussed. There are three main types of communication that can be assumed in an IoT network: unicast, group, and broadcast communication. Apart from that, Hamasaki et al. considered geometric characteristics and proposed a method of geometric group key sharing. Thereafter, a key sharing method suitable for sharing a pairwise key by implementing the method proposed by Hamasaki et al. had been proposed by Nishigami et al. However, testing this method, we found that when a node and its fellow nodes are attacked together, the keys of the rest of the nodes will be leaked. Therefore, in this paper, using the feature introduced in geometric group key sharing, we propose a method that enables a pairwise key to be securely shared. In addition, we extend our method of pairwise key sharing to be applicable for group key sharing to achieve a way to share efficiently pairwise, group, and global keys used in broadcast communication. Finally, we evaluate the efficiency of our proposed method.**

Keywords—Group Key Sharing, Pairwise Key Sharing, Global Key Sharing, Symmetric Key Encryption, IoT network

I. INTRODUCTION

In recent years, the concept of Internet of Things (IoT) network [11] enabling devices and everyday objects to communicate between each other has been extensively discussed. An IoT network generally comprises two types of devices: base station (BS) and nodes. Typically, only one BS exists within a network, and it acts as the central part of the network. In contrast, several nodes can exist in a single network collecting various types of information and sending them to the BS. The BS then utilizes information collected and performs analyzation and information processing. Owing to this, the BS is assumed to have the substantial amount of computational power. In addition, considering that the BS is assumed as the center of the network, in the case it fails to function properly, the network itself will also collapse. Therefore, the BS is generally assumed to be tamper-resistant, with the large amount of computational power, having abundant power source capacity, and secure. In contrast, nodes can sometimes be established in places that are physically unsafe. Moreover, as there are several nodes existing in a single network, it is desirable to minimize the cost of maintaining each node. Therefore, each node is assumed to be easily analyzed because it is not resistant towards tampering; moreover, each node is assumed to have low-performance central processing unit (CPU) with low power capacity.

Communication in an IoT network can be classified into three types of communication configuration as follows: unicast, group, and broadcast communication. Unicast communication is a one-to-one communication performed by two devices; and group communication is a communication when there are three or more devices connected together. Lastly, broadcast communication is a communication where all devices in the network are interconnected. This can also be viewed as a type of group communication where the "group" is defined as "all the nodes in the network".

In the aforementioned IoT network, to perform any type of communication securely, all communication processes need to be encrypted. To enable this, a secure method of key sharing corresponding to each type of communication configuration is essential. Therefore, considering the low performance of nodes as mentioned before, there are five criteria as shown below which are critically important for any key sharing method used in the IoT network:

- **Communication Cost**
  Taking into account the low power capacity of a node, it is important to ensure that a method used will have lower communication cost, as communication consumes the most of energy of a node.
- **Computation Cost**
  Because the CPU performance of a node is typically very low, a method with lower computation cost is desired.
- **Storage Cost**
  Performing key sharing, there is a method that shares a key through a combination of several preinstalled element keys. However, it is important to consider the case when the storage capacity of a node is low. Therefore, it is desirable that the amount of information needed to be stored in a node to be small.
- **Flexibility**
  Because nodes are not resistant to tampering, in the case when a key gets analyzed, if the same key is continuously used it means that all the information shared using this key afterwards will be leaked. Therefore, it is desirable that a key can be renewed easily. In addition, a mechanism of removing nodes is also required for the nodes that had leaked their keys.
- **Security**
  When one of nodes is successfully analyzed, it is important that the key used between nodes not related to the analyzed node is not revealed.

Considering the requirements mentioned above, Hamasaki et al. proposed a method [2, 3] of geometric group key sharing. This method employs the characteristics of geometric shapes of a circle or a sphere and has such feature as the easier key renewing process. In addition, evaluation of this method based on the aforementioned five criteria shows that it achieved an extremely good result. However, this method is not implementable in cases when the number of nodes is two, namely, in sharing pairwise key. This is because, in cases

when the number of nodes is two, construction of a circle or a sphere is impossible, instead, the two nodes will be connected by a straight line. Therefore, the problem is that having information on the center point and one of the node's coordinate point, the coordinate point of the other node can easily be specified. Thereafter, Nishigami et al. proposed an improved method of pairwise key sharing based on the geometric group key sharing [9]. However, in this research, we understand that in this method, when a node is compromised, the key for the rest of the nodes will be leaked as well. Therefore, in this study, using the feature of geometric group key sharing, we proposed a method that can share pairwise keys securely, and we evaluated the security of this new method. In addition, we proposed a method extending the proposed key sharing approach that is able to share a pairwise key, a group key, and a global key (the key used in broadcast communication) efficiently.

The remainder of this paper is organized as follows. In chapter II, we explain in detail several representative conventional methods of pairwise key and group key sharing. In chapter III, we show the method of Nishigami et al. and explain the problem appearing in this method. In chapter IV, we propose a secure geometric pair key sharing method and evaluate its performance. In addition, in chapter V, using the proposed pairwise key sharing method, we suggest the method that could enable sharing both group key and global key at the same time. Finally, in chapter VI, we conclude our results.

II. Conventional Methods

A. Localized encryption and authentication protocol [1,10]

In localized encryption and authentication protocol (LEAP), it is implied that each node will contain the following four keys. The first key is an individual key shared with the base station (BS), the second key is a group key shared between all the nodes in the network, the third key is a cluster key shared between multiple neighboring nodes, and the fourth key is a pairwise key shared through one-to-one communication between the neighboring nodes. Here, the group key is used in broadcasting, and the cluster key is used in multicasting. Among these four types of key, the individual key and the group key are prestored in the node.

The method of sharing pairwise keys is established as follows. First, an administrator produces one initial key and stores it in all nodes. Each node creates a master key by encrypting its own ID using the initial key. Then, each ID is exchanged with the neighboring node, and the master key is then used to encrypt the exchanged ID resulting in establishment of a pairwise key with the neighboring node. The neighboring node also performs the same process on the exchanged ID. However, the initial key is deleted after a set period of time. In contrast, to update the shared key, multiple master keys generated from multiple initial keys are stored beforehand and are used one by one in order. Therefore, the number of key updates is limited by the number of master keys prepared in advance.

In LEAP, because no BS is assumed, all processes and computations are performed by the node itself. The method of sharing group keys in LEAP is described below:

1. In the group of nodes that communicate between each other, one of the node acts as the Initiator (IN).
2. The IN shares the pairwise key with other nodes that compose the group by using the method mentioned before.
3. The IN generates the group key S, then encrypts it using the shared pairwise key and sends both the ID of the targeted node and the encrypted group key to the target node.
4. The node that received the encrypted group key from the IN decrypts it by using the pairwise key shared with the IN and obtains the group key S.

B. Hamasaki et al. method [2,3]

In Hamasaki et al. method, a BS is assumed, consequently, all nodes perform the process of key sharing through the BS. Let us assume that all nodes have their own unique individual key, and the BS knows each individual key of each node. In addition, all computations are performed using integer modulo p, and the number of nodes is n. The BS shares the group key according the procedure outlined below:

1. The BS generates a random number r, inputs the generated random number r and each node's individual key into the pseudorandom number generator (PRNG) and computes $n-1$ dimension coordinate points $a_i\left(a_{i1}, a_{i2}, \cdots a_{i(n-1)}\right)(i = 1,2, \cdots n)$ for each node.
2. If coordinates of the node are included in $(n-2)$-dimension plane, Step 1 is repeated by re-producing the random number r.
3. The BS computes the center point of the circle $o_i\left(o_{i1}, o_{i2}, \cdots, o_{i(n-1)}\right)$ $(i = 1,2,3, \cdots n)$ which passes through coordinates of each node.
4. The BS broadcasts the following information:

$$n, r, o_i\left(o_{i1}, o_{i2}, \cdots, o_{i(n-1)}\right)$$

Next, we explain in more detail the processes performed at i-th node:

1. First step is to input the random number r received from the BS and its own individual key into PRNG and to compute its own node coordinates $a_i\left(a_{i1}, a_{i2}, \cdots a_{i(n-1)}\right)$.
2. Second step is to compute the distance between the coordinate of the center point received from the BS to its own node coordinate a_i by using equation (1). However, as the objective for all nodes is to share the same value; therefore, the computation of square root is not performed.

$$S = (o_1 - a_{i1})^2 + (o_2 - a_{i2})^2 + \cdots + \left(o_{n-1} - a_{i(n-1)}\right)^2 \quad (1)$$

III. Pairwise Key Sharing by Nishigami et al. [9]

Let us suppose that there is a node that seeks to share a pairwise key with each of the surrounding nodes. In this case, the Hamasaki et al. method faces the following problem. If the number of nodes is two, a geometric shape of a circle or a sphere cannot be constructed, instead, the two nodes will be connected by a straight line. Therefore, there is a problem that having one of the node's coordinate and the coordinate of the center point, the coordinate of the other node can easily be specified. For example, let us suppose that node A seeks to share a pairwise key with surrounding nodes node B, node C, and node D. Node B having its own coordinate and the coordinate of the center point, can specify and determine the coordinate of node A. At this point, if the BS shares (broadcasts) the center point coordinate between node A and node C, node B can identify the pairwise key between node A

and node C. The same can be said with respect to the pairwise key of node A and node D. In addition, node C and node D can also identify the pairwise key between node A, and other nodes in the same way. Therefore, Hamasaki et al. method is not applicable to pairwise key sharing.

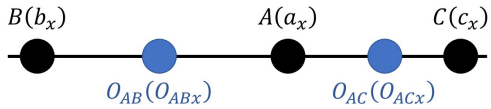

Fig. 1. Implementation of the Hamasaki et al. method into pairwise key sharing

A. Pairwise key sharing by Nishigami et al.

To solve the aforementioned problem, in addition to the coordinates of two nodes, Nishigami et al. proposed an approach of adding an imaginary point managed the BS, therefore, instead of a straight line, a geometric shape of a circle is formed, which allows sharing of key, as shown below.

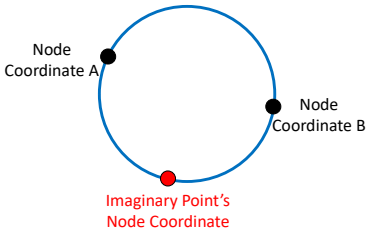

Fig. 2. Use of imaginary point

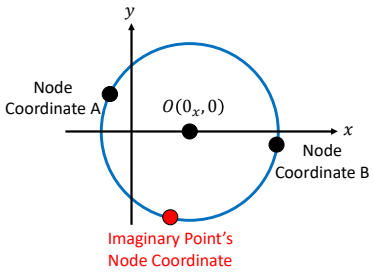

Fig. 3. Method for setting the imaginary point

Employing this approach, it is possible to guarantee that even if node B could specify the circumference of the circle using the center point, it cannot specify the exact location of node A based on that circumference. Therefore, even having information on the center point of node A and node C, node B will have to consider all possible coordinates on the circumference of the circle and all circles between that center point; therefore, revealing the pairwise key of node A and node C by node B is prevented.

In addition, to deal with the problem that the computational cost of the BS increases due to changes in the dimension of coordinate of the center point from one-dimensional point to two-dimensional point, Nishigami et al. derived a new way of setting the imaginary point as shown in Figure 3, where the center point is set in such way that it will lie on the x-axis. Consequently, the BS only needs to send the value of the x-axis's coordinate, thus achieving the same communication cost as in the case of one-dimension point.

In the method proposed by Nishigami et al., when node A shares a pairwise key with node B, unlike in the case of the aforementioned Hamasaki et al. method, node B cannot specify the coordinate of node A. Therefore, it is considered to be secure. However, as explained below, if node A shares the pairwise key with three or more nodes, there is a problem that in the case if two of the nodes are attacked, the pairwise key of the rest of nodes will also be leaked. For example, if both node B and node C are attacked considering that each of them had shared the pairwise key with node A, the coordinates of node A will be leaked as well. This happens because node B knows the coordinates of the center point o_1 and distance S1, and node C knows that of the center point o_2 and distance S2, consequently, solving the equations shown below they can identify the coordinate (a_1, a_2) of node A. In this case, if node B and node C know about the center point between node A and node D, they could also identify the pairwise key of node A and node D.

$$S1 = (o_1 - a_1)^2 + a_2^2 \quad (6)$$

$$S2 = (o_2 - a_1)^2 + a_2^2 \quad (7)$$

This is because the number of variables regarding node A is two. Therefore, node A needs to change its coordinate (a_1, a_2) each time it shares the pairwise key with a node. Typically, to update the position coordinate, for example, the random number r is changed each time, and the pseudorandom number is updated accordingly. However, in this case, because generation of pseudorandom number is performed according to the number of nodes, therefore, the efficiency of this method will be deteriorated.

IV. Proposed Method 1 (Pairwise Key Sharing)

To change the coordinate of node A easily, we perform the bit inversion. Below, we assume that node 0 shares a pairwise key with n number of surrounding nodes $i(i = 1, \dots, n)$. However, for simplicity, we assume that the key length L_2 is a prime number.

1. Node 0 seeking to share a pairwise key with node $i(i = 1, \dots, n)$ send its own ID and IDs of nodes $i(i = 1, \dots, n)$ to the BS.
2. The BS generates the random number r, inputs it along with each node's individual key into PRNG and finds the coordinates $(0_x, 0_y)$ and (i_x, i_y) of node 0 and node $i(i = 1, \dots, n)$, respectively.
3. In the case where $n \leqq L_2$, the BS generates the random number (l_1, q_1), then, each (l_1)-th bit and $(l_1 + i \times q_1)$-th bit of $(0_x, 0_y)$ are inverted producing $(0_{ix}, 0_{ix})$. For example, with regard to node 1, the value where the (l_1)-th bit and $(l_1 + q_1)$-th bit of $(0_x, 0_y)$ are inverted, is $(0_{1x}, 0_{1y})$. With regard to node 2, the value where the (l_1) -th bit and $(l_1 + 2q_1)$ -th bit are inverted, is $(0_{2x}, 0_{2y})$. When $i \times q_1 > L_2$, consequently, $i \times q_1 = i \times q_1 \bmod L_2$,
4. In the case where $(m-1)L_2 < n \leqq mL_2$, the BS generates the random number $(l_k, q_k)(k = 2, \dots, m)$, and each (l_k) -th bit and $(l_k + i \times q_k)$ -th bit of $(0_x, 0_y)$ are inverted producing $(0_{(k-1)L_2+ix}, 0_{(k-1)L_2+iy})$ for node $(k-1)L_2 + i$. For example, with regard to node $L_2 + 1$, concerning the new random number (l_2, q_2), the value where the (l_2)-th bit and $(l_2 + (L_2 + 1)q_2)$ -th bit are inverted, is $(0_{L_2+1x}, 0_{L_2+1y})$. However, if $n > L_2$, the value of l_k used is recorded, and combination of the same bit is avoided.

5. Regarding every i $(i = 1, \dots, n)$, the BS solves the following equation using the method proposed by Nishigami et al. and finds the value of the center point $O(o_{ix}, 0)$ so that it lies on the $x - axis$, consequently, it finds the value S_i.

$$S_i = (o_{ix} - 0_{ix})^2 + 0_{iy}^2 \quad (8)$$

$$S_i = (o_{ix} - i_x)^2 + i_y^2 \quad (9)$$

6. The BS then uses the key of node 0 to encrypt the value (l_k, q_k) $(k = 1, \dots, m)$ producing $Enc(l_k, q_k)$ and thereafter, broadcasts the following information.

$$r, ID_0, Enc(l_1, q_1), \dots, Enc(l_m, q_m),$$
$$(ID_1, o_{1x}), \dots, (ID_n, o_{n1})$$

7. Node i $(i = 1, \dots, n)$ inputs the value of r and its own individual key into the PRNG and finds its node coordinate (i_x, i_y).
8. Node i $(i = 1, \dots, n)$ substitutes its own coordinate and the center point into equation (9) and computes the pairwise key S_i.
9. Node 0 inputs the value of r and its own individual key into the PRNG and finds its node coordinate $(0_x, 0_y)$. Then, node 0 reconstructs $(l_1, q_1), \dots, (l_m, q_m)$ from $Enc(l_1, q_1), \dots, Enc(l_m, q_m)$.
10. Let us assume that the value where (l_k)-th bit and $(l_k + i \times q_k)$-th bit of node 0's coordinate $(0_x, 0_y)$ are inverted, is $(0_{ix}, 0_{ix})$. However, if $n > L_2$, the value of l_k used is recorded, and combination of the same bit is avoided.
11. Node 0 inputs each value of $(0_{ix}, 0_{ix})$ and o_{ix} into equation (8) and obtains the pairwise key S_i with each node i.

A. Evaluation of performance of the proposed method 1

In this section, we perform comparison for the considered methods in terms of computation cost, communication cost and storage cost for sharing pairwise key. Because the BS usually has enough computation capability and power supply capacity, evaluation with respect to the BS is not included. However, as $20\ bit \sim 30\ bit$ is usually enough for representing ID, we perform the evaluation considering that the parameter $L1 = 32\ bit$ and the length of the key $L2 = 127\ bit$. We assumed the cost for each broadcast as one time of communication. In addition, we perform our evaluation based on the assumption that computational cost of a single encryption and decryption process is $C2$, that of binomial is $C1$, and that of one bit inversion is $C0$. Generally, $C2 > C1 > C0$. In addition, we consider the case where node 0 (same as IN in LEAP) performs pairwise key sharing with the surrounding $n - 1$ number of nodes.

In LEAP, each node holds an individual key, and the IN requires n times of encryption $(nC2)$ and one time of communication for broadcasting ID $(L1)$. Nodes other than the IN require two times of encryption $(2C2)$ and one time of communication of ID $(L1)$.

In the proposed method 1, the IN performs one time of communication for broadcasting its own ID $(L1)$ and n times of communication for sending IDs $(nL1)$ to the BS. Nodes other than the IN perform one communication of ID $(L1)$. Therefore, the communication cost for the IN is $(n + 1)L1$, and communication cost for nodes other than the IN is $L1$. In addition, if $n \leqq L_2$, the IN performs two times of encryption and decryption, $2(n - 1)$ times of bit inversion process, and one computation of binomial. If $n > L_2$, the IN performs two times of encryption, $2m$ times of decryption, $2(n - 1)$ times of bit inversion process, and $(n - 1)$ times of computation of binomial. Therefore, total computational cost for the IN is $(2 + 2m)C2 + (n - 1)(2C0 + C1)$. In addition, nodes other than the IN perform two times of encryption and one computation of binomial, therefore, the computational cost is $2C2 + C1$. However, if $n > L_2$, the IN requires $L2$ bit of storage cost for recording the value of bits used, in addition to its own individual key. In other words, in the beginning, the $L2$ bit is filled with 0, when l_1 is used, $l_1\text{-}th$ bit of the $L2$ bit is changed to 1. When the $i\text{-}th$ bit in $L2$ is 1, consequently, use of the bit value corresponding to that position is avoided.

To present the results of comparison clearly, Table I represents them using only the dominant cost for computation and communication. For example, because $C2 > C1 > C0$, where the cost values of $C0$ and $C1$ are negligibly small in comparison to $C2$, in Table I we only show the computational cost related to $C2$. The same is done with regard to the communication that involves communication of $L1$ and $L2$, where $L2 > L1$. It should be noted that in Table I if $n \leqq L_2$, the value of $m = 0$, and storage cost is $L2$.

To conclude the performed comparison, we can say that our method requires the smallest computational cost.

TABLE I. PERFORMANCE EVALUATION OF PAIRWISE-KEY SHARING

	LEAP [1, 10]	Proposed Method 1
Communication Cost (IN)	$L1$	$(n + 1)L1$
Communication Cost (node)	$L1$	$L1$
Computation Cost (IN)	$nC2$	$(2 + 2m)C2$
Computation Cost (node)	$2C2$	$2C2$
Storage Cost (IN)	$L2$	$2L2$
Storage Cost (node)	$L2$	$L2$

B. Evaluation of security and flexibility of the proposed method 1

Next, we perform comparison with respect to security and flexibility when performing key sharing and key renewal.

First, we explain in detail the concepts of LEAP. As nodes are not resistant towards tampering, in the case of attacks such as stopping the timer of each node, the node is analyzed before it is able to eliminate its initial key, consequently, the initial key is leaked. If the initial key is leaked, there is a risk that the pairwise key for every node will also be leaked. In addition, as the number of possible key renewals in LEAP is limited, it is deemed to have low level of security and flexibility.

Furthermore, we discuss the security of our proposed method 1. In this case, the process of changing the coordinate is performed by inverting two different bits for each node. In the geometric key sharing method, the node's coordinate is not set as public. In addition, the place of inverting bits is sent after encrypting, therefore, the place of inverting bits for each node is also not set as public. Thus, let us suppose that node 0 performed pairwise key sharing with node 1, node 2, and node 3; both node 1 and node 2 conspired together and wanted to analyze node 3. Here, both node 1 and node 2 know the circumference of circle between themselves and node 0. However, as the coordinate of node 0 is different, there is no

common point of contact between these two circles. Therefore, for example, with regard to all coordinates on the circumference that node 1 is aware of, combination of two bits is inverted and if the inverted value exists on the circumference that is known by node 2, this point will become one possible candidate of coordinate 0. However, if the key length is made to be prime number of 127 bit, the total number of possible candidates is larger than 2^{127}, therefore, we can conclude that our proposed method is secure enough and could achieve a proper level of computational security.

In addition, as the bit inversion process is a nonlinear process, it cannot be uniquely represented by using basic four arithmetic operations. In general, bit decomposition is first performed, then the value of 1 is added per bit, and finally, computation of mod 2 is performed. Therefore, by using the difference between $(0_{ix}, 0_{iy})$ and $(0_{i+1x}, 0_{i+1y})$, and representing 0_{i+1x} and 0_{i+1y} as $0_{i+1x} = 0_{ix} + a_{ix}$, $0_{i+1y} = 0_{iy} + a_{iy}$, the equations can be formed as follows:

$$S_1 = (o_{1x} - 0_{1x})^2 + 0_{1y}^2 \quad (10)$$

$$S_2 = (o_{2x} - 0_{2x})^2 + 0_{2y}^2 = (o_{2x} - 0_{1x} - a_{1x})^2 + (0_{1y} + a_{1y})^2 \quad (11)$$

Node 1 and node 2 know the value of S_1, S_2, o_{1x}, o_{2x}, but without the knowledge of the difference a_{1x}, a_{1y}, equation shown above cannot be solved, therefore, node 1 and node 2 cannot identify the value of $0_{1x}, 0_{1y}$. In addition, even if the number of nodes conspired together increase, the number of both equations and variables a_{ix}, a_{iy} will also increase, therefore, we could say that the above argument remains valid and the equation cannot be solved.

In addition, if the value of $l_k + iq_k$ exceeds the value of the prime number $L2$, it is certain that it will return to a value that had not been used until now. This is due to the following reasoning. Let us suppose that the following relation of $(l_k + iq_k)\ mod\ L2 = (l_k + iq_k) - L2 = (l_k + jq_k)$ holds (j is a different value of i used until now). Moreover, combination of 2 bit out of $L2$ bit is ${}_{L2}C_2$ and the position of bit in this combination does not overlap. In other words, as there is no position of node 0 that is the same, even if nodes other than node 0 are attacked and analyzed, no information about the coordinate of node 0 can be revealed. Therefore, if the key length $L2 = 127$, the total possible number of combinations is ${}_{127}C_2$=8001, thus we can say that our method can handle up to $n = 8000$.

V. Proposed Method 2 (four keys sharing)

Considering key sharing in various situations, a method that allows sharing individual, pairwise, group, and global keys such as in LEAP is more desirable. Therefore, we proposed a method that could realize all the above, its main concept is discussed below.

A. Key sharing of pairwise key, group key, global key

We assume that an initiator (IN) is established in the same way as in LEAP; the IN shares a pairwise key with every surrounding nodes; the IN shares a group key with all surrounding nodes; and the IN shares a global key which is the same across the group. A concrete way for sharing pairwise key is shown in the proposed method 1. The IN shares the group key using the shared pairwise key. Every node has its own individual key, and the global key can be shared by applying the below method of group key sharing towards the whole group. Below, we show our proposed method 2. Here, with regard to a group $j(j = 1, \dots, l)$, ID of nodes that have the role of the Initiator (IN) is $ID_{j,0}$, and ID of surrounding nodes are $ID_{j,i}(i = 1, \dots, n_j)$. It should be noted that the IN knows all IDs of its surrounding nodes. The number of constructed groups is l.

1. Through the improved algorithm discussed in Section IV, node $ID_{j,0}(j = 1, \dots, l)$ shares a pairwise key $S_{j,i}$ with each of the surrounding nodes.
2. The BS encrypts a group key G_j by using $S_{j,i}$, and produces $g_{j,i}$.
3. The BS encrypts a global key B by using G_j, and produces b_j.
4. The BS broadcasts the following. However, it should be noted that $k = 1, \dots, m_j = \lceil n_j / L2_j \rceil$.
$$r, ID_{j,0}, e(l_{j,k}, q_{j,k}), (ID_{j,i}, o_{j,ix}, g_{j,i}), b_j$$
5. Using the center point $o_{j,ix}$ and its own coordinate, each node $ID_{j,i}$ generates its pairwise key $S_{j,i}$ with the IN, reconstructs the group key G_j from $g_{j,i}$ using $S_{j,i}$ and reconstructs the global key B from b_j using G_j.
6. Node $ID_{j,0}$ inputs a random number r and its own individual key into PRNG and computes $(0_x, 0_y)$, then it performs bit inversion as shown in Section IV.B producing the pairwise key $S_{j,i}$, thereafter, obtaining the group key G_j and the global key B through the same process as in Step 5.

B. Evaluation of performance of the proposed method 2

In this section, we perform comparison of the proposed method 2 allowing each node to hold individual, pairwise, group, and global keys against LEAP. In addition, for ease of understanding, we perform evaluation on a single group, and the number of nodes is equal to n. If all groups have the same number of nodes, the total cost will be a multiplication with l number of groups. However, for simplicity, we perform the evaluation under the setting that $n \leqq L_2$.

In LEAP, the IN requires $L1$ communication cost sharing a pairwise key and $(n-1)(L1 + L2)$ communication cost sharing a group key, therefore, the total communication cost is $nL1 + (n-1)L2$. In addition, the IN performs n times of encryption sharing a pairwise key and $(n-1)$ times of encryption sharing a group key. Therefore, the total computational cost of the IN is $(2n-1)C2$. On the other hand, nodes other than the IN perform $L1$ communication, two times of encryption and one time of decryption process resulting in total computational cost of $3C2$. In addition, each node is prestored with its own individual key. However, as the global key in LEAP is fixed, the process regarding global key is not required.

Next, we explain in detail our proposed method 2. In this case, each node has its own individual key, and nodes other than the IN send its own ID to the IN. The IN then broadcasts its own ID to the surrounding nodes once, and thereafter, sends all the IDs including IDs of the surrounding nodes to the BS.

In addition, in the proposed method 2, nodes other than the IN perform two times of encryption and two times of decryption. The IN performs two times of encryption and two

times of decryption (as the group key and the global key are the same for all nodes in the group, decryption of both keys is performed only once), as well as $2(n-1)$ times of bit inversion and $(n-1)$ times of binomial computation. Therefore, the IN requires a total of $4C2 + 2(n-1)(C1 + C0)$ computation cost.

TABLE II. PERFORMANCE EVALUATION OF PROPOSED METHOD 2

	LEAP [1, 10]	Proposed Method 2
Communication Cost (IN)	$(n-1)L2$	$(n+1)L1$
Communication Cost (node)	$L1$	$L1$
Computation Cost (IN)	$(2n-1)C2$	$4C2$
Computation Cost (node)	$3C2$	$4C2$
Storage Cost (IN)	$L2$	$L2$
Storage Cost (node)	$L2$	$L2$

Results of the comparison performed using only the dominant cost for computation and communication are summarized in Table II. It can be seen that in our proposed method, the computational cost for nodes other than the IN is higher than in LEAP, however, this is due to the process regarding the global key. As the global key in LEAP is fixed, the process related to global key is not required. Therefore, if we take the process regarding the global key out of the scope of the proposed method 2, computational cost for nodes other than the IN will be $3C2$, hence, equal to that of LEAP. However, if the global key in LEAP is renewed by using the same process as in the group key, cost for the IN will increase by $(n-1)L2$ for communication and $(n-1)C2$ for computation, consequently, the computation cost will be $4C2$, as nodes other than the IN require two times of encryption process.

Therefore, if the process regarding the global key is not taken into consideration, we can conclude that our proposed method 2 requires the lowest cost for both computation and communication.

C. *Evaluation on security and flexibility of the proposed method 2*

First, we perform comparison in terms of security. In LEAP, the security of group key sharing is directly linked to the security of the pairwise key. In addition, in our proposed method 2, as the pairwise key is the base for the group key, the security of the group key also depends on the security of the pairwise key. Therefore, to compare the security of these two methods, we only need to focus on the security of the pairwise key for both methods. As explained in Section IV.B, because our proposed method excels more in term of security, we can also say that our proposed method 2 also allows achieving better security for group key sharing.

Next, we compare the flexibility of each method. As explained in Section IV.B, in LEAP, the number of possible key renewals is limited. In contrast, in the proposed method 2, the number of key renewals is unlimited, and a key can be easily renewed when the BS sets a new random number r, therefore, our proposed method demonstrates better flexibility comparing to LEAP. In addition, in LEAP, as the global key is fixed, the global key will be leaked when one of the nodes is analyzed; however, in our proposed method, at the expense of a slight increase in the overall computational cost, the global key can be easily renewed.

VI. CONCLUSION

In this paper, on the base of the proposed method 1, we proposed a secure method of pairwise key sharing, and using the proposed method 2, we proposed the key sharing method that can perform sharing pairwise, group, and global keys. Both proposed methods were able to improve the efficiency of other conventional methods and provided a higher level of security and flexibility. Therefore, both of our proposed methods are suitable for application in Internet of Things (IoT) network, since both methods require low energy consumption while providing high flexibility and security at the same time.

REFERENCES

[1] Sencun Zhu, Sanjeev Setia, Sushil Jajodia: "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor networks." In Proceedings of the 10th ACM Conference on Computer and Communications Security, pp.62-72, New York, USA. (2003).

[2] Jun Hamasaki, Ryo Kaneko, Keiichi Iwamura: "Geometric Group Key-Sharing Scheme. " IEEE 5th Global Conference on Consumer Electronics (GCCE), pp.1-2. (2016).

[3] Jun Hamasaki, Keiichi Iwamura: "Geometric Group Key-Sharing Scheme using Euclidean Distance." 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp.1004-1005, Las Vegas, NV. (2017).

[4] Hidetoshi Yukimaru, Yoshio Kakizaki, Keiichi Iwamura: "Key management scheme applicable to various topologies of sensor networks." 6th International Conference on Availability, Reliability and Security, pp. 448-453, Vienna. (2011).

[5] Laurent Eschenauer, Virgil D. Gligor: "A Key Management Scheme for Distributed Sensor networks." In Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 41-47, New York, NY, USA. (2002).

[6] Haowen Chan, A. Perrig: "PIKE: Peer Intermediaries for Key Establishment in Sensor networks." Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 524-535, Miami, FL. (2005).

[7] Haowen Chan, A. Perrig, D. Song: "Random Key Predistribution Schemes for Sensor networks." Symposium on Security and Privacy, pp. 197-213, Berkeley, CA, USA. (2003).

[8] Ryo Kaneko, Kitahiro Kaneda, Keiichi Iwamura: "Comparative Evaluation of Large-Scale Wireless Sensor Network for Key Management.", IEICE Transactions on Information and Systems, Vol. J98-D, No. 3, pp. 418-427. (2015). (In Japanese)

[9] Koki Nishigami, Keiichi Iwamura: "Geometric pairwise key-sharing scheme." In: Lanet JL., Toma C. (eds) Innovative Security Solutions for Information Technology and Communications (SECITC 2018). Lecture Notes in Computer Science, vol 11359, pp. 518-528. Springer, Cham. (2018).

[10] Sencun Zhu, Sanjeev Setia, Sushil Jajodia: "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks." ACM Transactions on Sensor networks (TOSN), Vol. 2, Issue 4, pp. 500-528, New York, USA. (2006).

[11] Yiying Zhang, Chunying Wu, Jinping Cao, Xiangzhen, Li, "A Secret Sharing-Based Key Management in Hierarchical Wireless Sensor Network.", In International Journal of Distributed Sensor Networks, June 2013. (2013)

[12] Pawani Porambage, An Braeken, Corinna Schmitt, Andrei Gurtov, Mika Ylianttila, Burkhard Stiller: "Group Key Establishment for Enabling Secure Multicast Communication in Wireless Sensor Networks Deployed for IoT Applications." in IEEE Access, Vol. 3, pp. 1503-1511. (2015).

[13] W. Diffie, M. Hellman: "New Directions in Cryptography.", IEEE Transactions on Information Theory, Vol. 22, Issue 6, pp. 644-654. (1976).

[14] Wenliang Du, Jing Deng, Yunghsiang S. Han, Pramod K. Varshney, Jonathan Katz, Aram Khalili: "A Pairwise Key Predistribution Scheme for Wireless Sensor networks." ACM Transactions on Information and System Security (TISSEC), Vol. 8, Issue 2, pp. 228-258. (2005).

[15] Yiying Zhang, Chunying Wu, Jinping Cao, Xiangzhen Li: "A Secret Sharing-Based Key Management in Hierarchical Wireless Sensor Network." In International Journal of Distributed Sensor Networks, Vol. 9, Issue 6. (2013).