Volume: 14 Issue: 03 Pages: 5493-5498(2022) ISSN: 0975-0290

A Modified Hierarchical Multiple Key Agreement Scheme for WSN

Reza Alimoradi
Department of Computer Science, Faculty of Science, Qom University, Qom, Iran
Email: r.alimoradi@qom.ac.ir
Fateme Amjadi
Department of Computer Science, Faculty of Science, Qom University, Qom, Iran
Email: Amjadi.f.h@gmail.com
Seiied-Mohammad-Javad Razavian
Department of Computer Science, Faculty of Science, Qom University, Qom, Iran
Email: mjrazavian@qom.ac.ir
M. H. Noorallahzadeh
Faculty of Mathematical Sciences, Qom University, Qom, Iran
Emlail: Mh.noorallahzadeh@stu.qom.ac.ir

ABSTRACT

Nowadays, sensor networks are one of the hottest scientific issues. A lot of research has been done to improve their efficiency. Wireless Sensor Networks (WSN) are applied as an important and efficient technology in many industries such as military operations, security systems, intelligent transportation systems, medics, agriculture, and many others. Key agreement is a challenging point in the security of these networks. Sensor nodes connect to each other using cryptography techniques, however, use of the classic key management techniques such as key distribution center is inefficient because of resource-constrained nature of the sensor nodes. This paper proposes a hierarchical multiple key agreement scheme. In the proposed scheme, two nodes can produce multiple session keys, just with only one run of the key agreement protocol by two nodes in the hierarchical system. As well as its efficiency, this new scheme is based on identity and non-interactive protocol. Being zero-knowledge proof is another advantage of the scheme.

Keywords - **WSN, Key management, Hierarchical multiple key management, Public key cryptography, Multiple key agree-ment, Identity-based encryption**.

Date of Submission: Jul 31, 2022 Date of Acceptance: Nov 07, 2022

I. INTRODUCTION

Miniaturization and development of computational instru-ments in wireless network, makes a new type of computer network like WSN [1]. Developments in electronics and wire-less communications allow cheaper multi-purpose sensors with low energy consumption. The small sensor nodes equipped with short-range radio communication, sensing, and data processing, which run on battery [2, 3]. Today, the WSNs are used in traffic monitoring, pipeline monitoring, landslide diagnosis, methane leak detection, border patrol, precision agriculture, health care and rehabilitation applications, education, asset tracking, real-time monitoring of football (real-time soccer playing monitoring), fire control, water quality monitoring, martial applications, tracking military targets, penetration de-tection, authentication, household automation and trade industries [1, 4, 5]. Hierarchical networks have many advantages than the flat networks, including more system power, less system latency, and more system saving [4, 6]. The hierarchal sensor networks consist of multiple levels including sensor nodes, cluster head nodes, key distribution center node.

Security services like authentication and key management are vital for a secure connection between the nodes in adver-sarial environments. One of the most essential security services is producing a pairwise key which enables the sensor nodes to connect each other's using cryptography techniques [7]. How-ever, classic pairwise key production techniques like public key cryptography and key distribution center are not applicable because of resource limitations in sensor nodes [1, 8].

Gennaro et al. proposed an efficient and non-interactive hi-erarchical key agreement protocol which is suitable for mobile ad-hoc networks [9]. Their protocol is a pairing-based cryptography. Gennaro et al.'s proposed protocol is not applicable to all type of the WSNs because of its special design. Then, Kim introduced two non-interactive hierarchal key agreement protocols for the WSNs named Naïve Hierarchical Key Agreement Protocol (HKAP) and Privacy HKAP which both were revised version of Gennaro et al.'s scheme [7]. Kim's protocol did not support the freshness of the established session key, while a key agreement protocol must support this feature. In order to solve this problem, Lee and Kim provided two protocols. The first one is a naïve HKAP supporting features such as non-interactive, hierarchal, flexible and freshness of the established session keys. The second scheme is the privacy HKAP which its implementation is

Volume: 14 Issue: 03 Pages: 5493-5498(2022) ISSN: 0975-0290

based on naïve HKAP and it supports freshness of the established session key and anonymity.

Both protocols run in two phases: 1) Hierarchal key set-tlement phase and 2) session key agreement and secure com-munication phase. The former is for system regulation and the latter is for making a secure communication channel after session key agreement between every two nodes in the WSN. These two new protocols support security, power and freshness of session key in the hierarchal WSN [8]. Key management is a core mechanism to make sure that network services and utilities in WSNs are secure. The goal of key management in WSNs is solving the problems of creating, distributing, and maintaining the private keys [10]. In this paper, we propose a hierarchal multiple key agreement protocol. Key agreement schemes are one of the important issues in cryptography and they are used for producing a secret common key between two parts of an insecure network. Since multiple key agreement schemes can agree on more keys compared with single-key agreement schemes, then they are more efficient. Chaturvedi et al. [14] proposed a new key agreement protocol. They said that when we talk about modern efficient computers, the vulnerability of existing key agreement schemes increases further, so they proposed a key agreement protocol that works in a non-revolving group. The preliminaries go on in section 2. Lee and Kim's hierarchal key agreement protocol is reviewed in section 3. Section 4 presents the paper's suggested protocol. Computational complexity of the protocol is covered in section 5. Finally, section 6 deals with the security analysis of the protocol.

II. PRELIMINARIES

Like other identity-based key agreement identification protocols mentioned in [12-13], Lee's naïve HKAP needs a private key producer which includes 2 phases: key management, session key agreement and secure communication phase.

Let k be the security parameter, G and G_T be two cyclic group of the order q and $e : G * G \rightarrow G_T$ be a bilinear pairing. Using G^* Kim marks the set of non-identity G parameters.

Kim also assumes that the public keys (IDs or identities) which are in the depth of L are vectors of $(G^*)^L$ elements and the j-th element equals j's level identity. This system, then develops structures of public keys over $\{0,1\}^*$ by hashing each element of I_j using hash function resistant against clash $H:\{0,1\}^* \rightarrow G$. Abbreviations used in this protocol are in table 1.

Table 1. Notations

Notations	Description
$\boldsymbol{CH_i}$	Cluster head $\boldsymbol{i}$
$\boldsymbol{CM_{ij}}$	Member node *j* in the cluster head $\boldsymbol{i}$
$\boldsymbol{ID_i}$	Entity *i*'s identifier
$\boldsymbol{Q_i}$	Amplified identity of $\boldsymbol{ID_i}$
$(\boldsymbol{S_1, S_2, S_3})$	Private key set of sinks, $\boldsymbol{Si} \in \boldsymbol{Zq^*}$
$\boldsymbol{sk}$	Session key established between two entities
$\boldsymbol{R_1}$	Random number generated by $\boldsymbol{CM_{ij}}$
$\boldsymbol{G, G_T}$	Cyclic groups of prime order q
$\boldsymbol{P}$	Denote a generator of $\boldsymbol{G}$
$\hat{e}$	bilinear map $\boldsymbol{G \times G \rightarrow G_T}$
$\boldsymbol{H, H_1}$	One way hash function $\boldsymbol{h}: \{\boldsymbol{0,1}\}^* \rightarrow \boldsymbol{G^*}$
.	Multiplication
||	Concatenation
a, b	Temporary private keys generated by $\boldsymbol{CM_{ij}}$ and $\boldsymbol{CM_{kl}}$

III. LEE AND KIM'S HIERARCHAL KEY AGREEMENT PROTOCOL WITH FRESH KEY

The key agreement protocol between the participants must guarantee that every shared session key is fresh and will not be used again by one of the participants. Also, it means that the key used in a cryptographic sharing must not be used in another sharing. Therefore, it is necessary that the session key be changed continuously, as it may be at risk before use or at operation phases.

By Kim's naïve HKAP, in each section, the sk between every two entities will be computed which depends on both the private key and their multiple identities. In contrast for the session, this sk depends on a random value. Hence the naïve HKAP does not support the freshness of the session key. Lack of support for freshness means that generated keys in different sessions are always the same which can provide the intruder useful information.

Lee and Kim proposed two hierarchal key agreement protocols with freshness: naïve HKAP FP and private HKAP FP. Both of them run in two phases: 1) hierarchal key management phase, and 2) key agreement and secure communication phase. The first phase is for system regulation like the same phase in Kim's protocol and the second phase is for making a connection through a secure channel after setting a fresh key between every two nodes in hierarchal WSN.

A. The Naïve HKAP-FP:

In order to produce a common key between two nodes in WSN, some secret keys must be produced beforehand. The goal of the key management phase is producing the required secure keys before their development. In fact, in WSNs, the nodes are met before development. This is a major difference between the WSN environments and mobile network environments.

a) The Hierarchal key management phase: This phase allows each node to have a pair of keys for public key cryptography; one public key and one private key.

Volume: 14 Issue: 03 Pages: 5493-5498(2022) ISSN: 0975-0290

The Sink node plays the role of PKG. It is assumed that the Sink node is stronger than other nodes, cluster heads are superior over sensor nodes but they are inferior to the Sink nodes. The sensor nodes have less rights compared with other nodes. The nodes' roles are defined before this phase. To manage the keys this protocol follows these steps:

Step1: Sink with ID_S produces the set of private keys (S_1, S_2, S_3) for WSN and computes $ID_S.S_1$. Then saves the data in the memory and sends $\{(ID_S.S_1,\ S_2,\ S_3), ID_S\}$ for the cluster heads.

Step 2: When the cluster head with ID_{CH_i} receives the message, computes $ID_{CH_i}.S_2$ and saves this data in the memory and sends $\{(ID_S.S_1, ID_{CH_i}.S_2, S_3), ID_S, ID_{CH_i}\}$ to its subset nodes.

Step 3: When the sensor node with $ID_{CM_{ij}}$ receives the message, computes $ID_{CM_{ij}}.S_3$ and saves in its memory.

b) The Key Agreement and Secure Connection Phase: The aim of this phase is making a secure channel between each two nodes by producing a fresh session key. In order to generate a session key these steps must be taken:

Step 1: CM_{ij} with the private key set $(ID_S.S_1, ID_{CH_i}.S_2, ID_{CM_{ij}}.S_3)$ chooses a random value for r_1 and computes $R_1 = r_1.ID_{CM_{ij}}$.and generates the $sk = e(ID_S.S_1, ID_{S'}).e(ID_{CH_i}.S_1, ID_{CH_{k'}}).e(ID_{CM_{ij}}.S_3, ID_{CM_{kl'}})^{r_1}$ key using the set of ID corresponding with the counterpart node CM_{kl} which is represented as $\{ID_{S'}, ID_{CH_{k'}}, ID_{CM_{kl'}}\}$ then computes $MAC_1 = h(sk, R_1)$ and sends $\{R_1, MAC_1\}$ to CM_{kl}.

Step 2: When CM_{kl} receives the message, with the private key set $(ID_S.S_1, ID_{CH_k}.S_2, ID_{CM_{kl}}.S_3)$ and using the ID set corresponding with the facing node computes this fresh session key: $sk = e(ID_S.S_1, ID_{S'}).e(ID_{CH_k}.S_1, ID_{CH_{i'}}).e(ID_{CM_{kl}}.S_3, R_1).$

CM_{ij} and CM_{kl} can produce similar fresh session keys because

$$sk = e(ID_S.S_1, ID_S).e\big(ID_{CH_i}.S_2, ID_{CH_k}\big).e(ID_{CM_{ij}}.S_3, ID_{CM_{kl}})^{r_1} = e(ID_S.S_1, ID_S).e\big(ID_{CH_i}.S_2, ID_{CH_k}\big).e(ID_{CM_{ij}}.S_3, R_1) = e(ID_S, ID_S)^{S_1}.e(ID_{CH_k}, ID_{CH_i})^{S_2}.e(ID_{CM_{kl}}, ID_{CM_{ij}})^{S_3 r_1} = e(ID_S, ID_S)^{S_1}.e(ID_{CH_i}, ID_{CH_k})^{S_2}.e(ID_{CM_{ij}}, ID_{CM_{kl}})^{S_3 r_1}$$

CM_{kl} trusts the generated session key only if the comparison of MAC_1 with $h(sk, R_1)$ has been proved valid.

Step 3: CM_{kl} sends the encrypted data package with this message $MAC_2 = h(sk||the\ encrypted\ data\ packet)$ to the facing node CM_{ij} which has been encrypted by the agreed on session key SK.

Step 4: When the message is received, CM_{ij} checks the validity of MAC_2 with the agreed on session key Sk, only if this checks the validity proved successful, CM_{ij} accepts the message from CM_{kl} which means the encrypted message has been successfully transmitted through the channel secured with the Sk.

B. 2-3- The Private HKAP_FP:

In order to support the private issue, Lee et al. represented a private HKAP_FP based on the naïve model which supports the privacy using the nodes' corresponding ID_S not their real ID_S.

a) ***The Hierarchal Key Agreement Phases:*** *The steps and suppositions of this phase are identical with those of Kim's Private HKAP figure 1 shows the Hierarchal key management phase for the private HKAP_FP to manage the keys these actions must the done in this phase*

Step 1: Sink with identities ID_S generates a set of private keys as (S_1, S_2, S_3) for the WSN and computes the value of $Q_S = h(ID_S)$ and $Q_S.S_1$ while $h()$ is a one-way hash function. Then Sink saves these data in its memory and sends $\{(Q_S.S_1,\ S_2\ ,\ S_3),\ Q_S\}$ to head clusters.

Step 2: When the cluster head with identities ID_{CH_i} receives the message, it computes the value of $Q_{CH_i} = h(ID_{CH_i})$ and $Q_{CH_i}.S_2$ and then saves the data to its memory and sends $\{(Q_S.S_1, Q_{CH_i}.S_2, S_3), Q_S, Q_{CH_i}\}$ to the nods that are members of its cluster.

Step 3: When the sensor nod with the identity $ID_{CM_{ij}}$ receiving the message, it computes the value of $Q_{CM_{ij}} = h(ID_{CM_{ij}})$ and $Q_{CM_{ij}}.S_3$ and then saves these data $\{(Q_S.S_1, Q_{CH_i}.S_2, Q_{CM_{ij}}S_3), Q_S, Q_{CH_i}, Q_{CM_{ij}}\}$ to its memory.

b) *The Session Key Agreement and Secure Connection Phase: Tis phase acts similar to the naïve model. To generate a common fresh key CM_{ij} and CM_{kl} do the following steps:*

Step 1: CM_{ij} with the private key set $(Q_S.S_1, Q_{CH_i}.S_2, Q_{CM_{ij}}.S_3)$ chooses the random number r_1, computes the value of $R_1 = r_1.Q_{CM_{ij}}$ using the facing node's corresponding ID set CM_{kl} generates the fresh key $sk = e(Q_S.S_1, Q_{S'}).e(Q_{CH_i}.S_2, Q_{CH_{k'}}).e(Q_{CM_{ij}}.S_3, Q_{CM_{kl'}})^{r_1}$ which is represented as $\{Q_{S'}, Q_{CH_{k'}}, Q_{CM_{kl'}}\}$, then it computes the value of $MAC_1 = h(sk, R_1)$ and sends $\{R_1, MAC_1\}$ to CM_{kl}.

Step 2: When CM_{ij} receives the message, with the private key set $(Q_S.S_1, Q_{CH_k}.S_2, Q_{CM_{kl}}.S_3)$ and using the facing nod's corresponding ID set computes $sk = e(ID_S.S_1, ID_{S'}).e(ID_{CH_k}.S_1, ID_{CH_{i'}}).e(ID_{CM_{kl}}.S_3, R_1).$

CM_{ij} and CM_{kl} can produce identical fresh session keys because

$$sk = e(Q_S.S_1, Q_S).e(Q_{CH_i}.S_2, Q_{CH_k}).e(Q_{CM_{ij}}.S_3, P_{CM_{kl}})^{r_1} = e(Q_S.S_1, Q_S).e\big(Q_{CH_k}.S_2, Q_{CH_i}\big).e\big(Q_{CM_{kl}}.S_3, R_1\big) =$$

Volume: 14 Issue: 03 Pages: 5493-5498(2022) ISSN: 0975-0290

$e(Q_S, Q_S)^{S_1}.e(Q_{CH_k}\ Q_{CH_i})^{S_2}.e(Q_{CM_{kl}}, Q_{CM_{ij}})^{S_3 r_1} =$
$e(Q_S, Q_S)^{S_1}.e(Q_{CH_i}\ Q_{CH_k})^{S_2}.e(Q_{CM_{ij}}, Q_{CM_{kl}})^{S_3 r_1}$

CM_{kl} ensures the rectitude of the fresh session key only if proving the comparison of MAC_1 with $h(SK, R_1)$ is successfully done.

Step 3: CM_{kl} sends an encrypted data package along with the message $MAC_2 = h(sk \,||\, the\ encrypted\ data\ packet)$ to the facing node CM_{ij} which is encrypted by the agreed on session key sk.

Step 4: When the encrypted message is received, CM_{ij} proves the authenticity of MAC_2 using the agreed on fresh session key. If this was successful, CM_{ij} accepts the message from CM_{kl} which means the encrypted message has been successfully transmitted through the channel secured by SK.

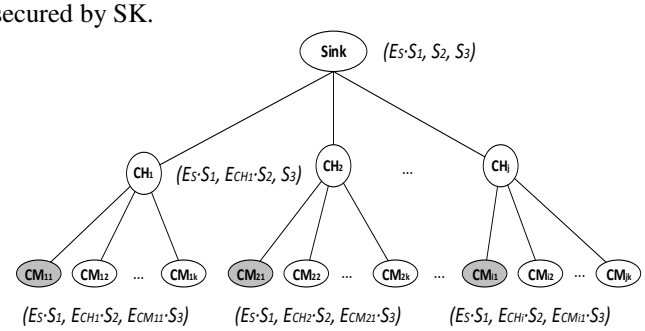

Fig. 1: The Hierarchal Key Management model for the private HKAP_FP.

IV. PROPOSED PROTOCOL

This section present a hierarchal multiple-key agreement protocol featuring non-cooperativeness, freshness and more security in WSN. This protocol runs in two phases: The first one is hierarchal key management and the second one is key agreement and secure connection. Similar to Lee and Kim, we assume it is a hierarchal network and there is a hop between sensor nodes and each cluster heads. Also, there are some hops between the cluster heads and Sink. Thus, in this paper regarding the hierarchal WSN, a hierarchal tree with the depth of 3 is assumed. We assume the degree of the Sink node is *u* and the degree of cluster head is *v*. This protocol uses the previous schemes to create equally distributed clusters in the network.

In the following, we will generate two common session keys instead of only one using multiple-key agreement. This will be possible with adding S_4 to the private key set produced by the Sink.

The proposed private protocol runs in the two following
phases:

a) The hierarchal key management phase: Assumptions and steps of this phase for the proposed private protocol are similar to those ones of Lee and Kim's private HKAP FP Step 1: Sink with ID_S creates a private key set of (S_1, S_2, S_3, S_4) for WSN and computer the value of $Q_S = h(ID_S)$ and $Q_S.S_1$ in which $h(\,)$ is a one-way and sends $\{(Q_S.S_1,\ S_2\ , S_3, S_4),\ Q_S\}$ to the cluster heads.

Step 2: The cluster head with the ID_{CH_i} received the message. Then computes the value of $Q_{CH_i} = h(ID_{CH_i})$ and $Q_{CH_i}.S_2$, and saves the data to this memory and sends $\{(Q_S.S_1, Q_{CH_i}.S_2, S_3, S_4), Q_S, Q_{CH_i}\}$ to the nodes which, members of its cluster.

Step 3: After the sensor node with $ID_{CM_{ij}}$ receives the message, it computes it value of $Q_{CM_{ij}} = h(ID_{CM_{ij}})$ and $Q_{CM_{ij}}.S_3$ saves this data $\{(Q_S.S_1, Q_{CH_i}.S_2, Q_{CM_{ij}}S_3, S_4), Q_S, Q_{CH_i}, Q_{CM_{ij}}\}$ to its memory.

b) *Key Agreement and secure connection phase:* The purpose of this stage is creating a secure channel by generating fresh keys between each two nods in the WSN. To produce common fresh keys CM_{ij} and CM_{kl} do as follows

Step 1: CM_{ij} with the private key set $(Q_S.S_1, Q_{CH_i}.S_2, Q_{CM_{ij}}.S_3, S_4)$ chooses the random number r_1 and computes $R_1 = r_1.Q_{CM_{ij}}$. Then, using the counterpart node corresponding ID set creates fresh session keys $sk1 = e(Q_S.S_1, Q_{S'}).e(Q_{CH_i}.S_2, Q_{CH_{k'}}).e(Q_{CM_{ij}}.S_3, Q_{CM_{kl'}})^{r_1}$ a, computes $MAC_1 = h(sk1, sk2, R_1)$ and sends $\{R_1, MAC_1\}$ to CM_{kl}.

Step 2: When CM_{kl} receives the message, with the private key set $(Q_S.S_1, Q_{CH_k}.S_2, Q_{CM_{kl}}.S_3, S_4)$ and using the counterpart node corresponding ID set computes these fresh keys

$sk1 =$
$e(Q_S.S_1,\ Q_{S'}).e(Q_{CH_k}.S_2, Q_{CH_{i'}}).e(Q_{CM_{kl}}.S_3, R_1)$
$sk2 =$
$e(Q_S.S_1,\ Q_{S'}).e(Q_{CH_k}.S_2, Q_{CH_{i'}}).e(Q_{CM_{kl}}.S_3, S_4.R_1)$

CM_{ij} and CM_{kl} can produce identical fresh session keys because

$sk1 =$
$e(Q_S.S_1,\ Q_S).e(Q_{CH_i}.S_2, Q_{CH_k}).e(Q_{CM_{ij}}.S_3, Q_{CM_{kl}})^{r_1} =$
$e(Q_S.S_1,\ Q_S).e(Q_{CH_k}.S_2, Q_{CH_i}).e(Q_{CM_{kl}}.S_3, R_1) =$
$e(Q_S,\ Q_S)^{S_1}.e(Q_{CH_k}, Q_{CH_i})^{S_2}.e(Q_{CM_{kl}}, Q_{CM_{ij}})^{S_3 r_1} =$
$e(Q_S,\ Q_S)^{S_1}.e(Q_{CH_i}, Q_{CH_k})^{S_2}.e(Q_{CM_{ij}}, Q_{CM_{kl}})^{S_3 r_1}$

$sk2 =$
$e(Q_S.S_1,\ Q_S).e(Q_{CH_i}.S_2, Q_{CH_k}).e(Q_{CM_{ij}}.S_3, S_4.Q_{CM_{kl}})^{r_1}$
$=$
$e(Q_S.S_1,\ Q_S).e(Q_{CH_k}.S_2, Q_{CH_i}).e(Q_{CM_{kl}}.S_3, S_4.R_1) =$
$e(Q_S,\ Q_S)^{S_1}.e(Q_{CH_k}, Q_{CH_i})^{S_2}.e(Q_{CM_{kl}}, Q_{CM_{ij}})^{S_3 S_4 r_1} =$
$e(Q_S,\ Q_S)^{S_1}.e(Q_{CH_i}, Q_{CH_k})^{S_2}.e(Q_{CM_{ij}}, Q_{CM_{kl}})^{S_3 S_4 r_1}$

CM_{kl} makes sure of the correctness of the fresh session key only and only if the check the validity of between MAC_1 and $h(sk1, sk2, R_1)$ is successful.

Step 3: CM_{kl} sends an encrypted data packet with this message

$MAC_2 =$
$h(ID_{CM_{ij}} || ID_{CM_{kl}} || sk1 || sk2 || the\ encrypted\ data\ packet)$

Volume: 14 Issue: 03 Pages: 5493-5498(2022) ISSN: 0975-0290

to the facing node CM_{ij} which is itself encrypted by the agreed session keys $sk1$ and $sk2$.

Step 4: After receiving the encrypted message, CM_{ij} check the authenticity of MAC_2 using the agreed on fresh session keys. Only if this stage is successful, CM_{ij} accepts the message from CM_{kl} which means the encrypted message has been successfully transmitted from the secure channel.

A. Computational Complexity

B. Security Analysis

Bilinear graph plays an important role in significant cryptography problems like in Bilinear Diffie-Hellman (BDH) problem which was presented by Boneh and Franklin and explained in [13]. Security of our proposed protocol is based on a type of this assumption. Assume G and G_T as cyclic groups of the order q and $e: G \times G \rightarrow G_T$ is a bilinear graph consider these computational assumptions:

- BDH: for each $a \in_R Z_q^*$, b and c and considering the given aP, bP and cP, computing $(P,P)^{abc}$ is difficult.
- Decisional Bilinear Diffie-Hellman (DBDH): for each $a, b \text{ and } c \in_R Z_q^*$, differentiating $(aP, bP, cP, e(P,P)^{abc})$ and $(aP, bP, cP, e(P,P)^r)$ is difficulty.

Theorem 1: The present scheme gives entity anonymity.

Proof: In this protocol, the anonymity of entity is found by the hash function and the BDH problem. The key management phase, secure connection and key agreement phase are done through each identity's corresponding ID and hashing its real ID, Only Sink has access to real ID_S of cluster heads and nodes. Even if the intruder finds the transmitted messages in the secure connection and key agreement phase he can never induce the real ID_S.

Theorem 2: The present protocol cannot disclose the agreed on session keys.

Proof: Consider the confidentiality of the private key set. The key set is a combination of the corresponding ID_S with the encrypted amount. It means that the intruder must know both data to get access to the private key set.

But even if the intruder hacks Sink, there's no way for him to find the two data. Also, in order to get the session keys, the intruder must try to induce $sk1$ and $sk2$ from each $\{R_1, MAC_1\}$ and {an encrypted data packet, MAC_2} the encrypted data packet. But there is no way for him to hack any entity.

Theorem 3: The present protocol supports the freshness of the session key and as a result can avoid the replay attack.

Proof: The freshness in the secure connection and key agreement phase means it guarantees the freshness of the session key. To achieve freshness, we use an r_1 with MAC_1 to generate session keys $sk1$ and $sk2$. But because of the BDH problem, there's no possible way for the intruder to generate session keys but the intruder has no way to generate session keys because of the BDH even if he could hack an entity. Therefore, because of the freshness of session keys the proposed protocol is immune against the replay attack.

Theorem 4: The proposed scheme is immune against non-passive intruder.

Proof: Assume that the intruder is successful only if he can get some useful data from the transmitted messages. We show that the possibility of his learning them is negligible because of the difficulty of the basic cryptography system, BDH problem and the DBDH problem.

1. Completeness of the present key agreement scheme has been verified in section 4.
2. If the enemy is a non-passive intruder, all intruders can get access to the corresponding ID set $\{Q_{S'}, Q_{CH_{ij}'}, Q_{CM_{ij}'}\}$ and the forwarded message MAC. But because of the difficulty of the basic cryptography sys, BDH problem and the DBDH. The probability of getting the information relate of the key from them is meagre. Finally, the present protocol is immune against non-active attacks.

Theorem 5: The suggested protocol is secure against action attacks.

Proof: Suppose the intruder is successful only if he finds the session keys $sk1$, $sk2$ and the information related to the session key $\{S_1, S_2, S_3, S_4\}$. Now, like the reasons of the previous theorem we will show that the possibility of success is meagre.

1. Acceptance by any entity means each MAC in the corresponding is successfully verified. That is MAC has been successfully encrypted and verified by the session key. We show that only in this case the entities accept the message and continue the session. Therefore, the probability of the enemy's changing the transmitted message is meagre. The only way to find session keys or their related information is solving difficulty of the basic cryptography system, the BDH problem and DBDH problem.
2. Now consider the active intruder as follows.
   a) There's no way for the enemy to get the encrypted data $\{S_1, S_2, S_3, S_4\}$ resulted from the difficulty of DBH problem.
   b) The intruder cannot cheat each CM_{ij} or CH_i by forging Sink's identity. As explained above, only the authentic Sink can from a legal message which includes an appropriate control that needs to be checked by the information of CM_{ij} and CM_{kl}. Even if the intruder was able to get verified at the stages of the protocol, still he still cannot get any useful information from the encrypted message. That is because of the difficulty of the basic cryptography sys, not producing authentic and resulting message.

Finally, can say this protocol is secure against active attacks.

Volume: 14 Issue: 03 Pages: 5493-5498(2022) ISSN: 0975-0290

Theorem 6: The presented protocol has the property of zero-knowledge proof. It means that the identifier has been successfully identified himself to the verifier without disclosing any of his secret data.
Proof: The CM_{kl} has this information $\{Q_{CM_{ij}}, Q_{CH_i}, Q_S, R_1\}$ from the other party. Finding the private value of CM_{ij} that is r_1 from $R_1 = r_1.Q_{CM_{ij}}$ is impossible because h is one-way and the difficulty of the discrete logarithm problem. Therefore, this protocol is zero-knowledge proof.

V. CONCLUSION

The scheme proposed in this article is a hierarchal multiple key agreement protocol which is the result of adding the minimum value S_4 to these sent at the key management phase in Lee et al.'s protocol, two session keys are produced has time complexity and less computations comparison with Lee et al.'s scheme where for producing two session keys the protocol needed to be run twice.

REFERENCES

[1] H. Kim, "Non-Interactive Hierarchical Key Agreement Protocol Over Hierarchical Wireless Sensor Networks," in Computer Applications for Security, Control and Sys-tem Engineering. Springer, 2012, pp. 86–93.

[2] Yong Wang and Garhan Attebury and Byrav Rama-murthy, "A survey of security issues in wireless sensor networks," IEEE Communications Surveys & Tutorials, vol. 8, pp. 2–23, 2006.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Com-munications magazine, vol. 40, no. 8, pp. 102–114, 2002.

[4] J. P. Yick, "Advanced Services in Wireless Sensor Net-works," Ph.D. dissertation, Davis, CA, USA, 2007.

[5] M. Turkanovic, B. Brumen, and M. Holbl, "A Novel User Authentication and Key Agreement Scheme for Heterogeneous Ad Hoc Wireless Sensor Networks, Based on the Internet of Things Notion," vol. 20, p. 96–112, 04 2014.

[6] L. B. Oliveira, H. C. Wang, and A. A. Loureiro, "LHA-SP: Secure Protocols for Hierarchical Wireless Sensor Networks," in Integrated Network Management, 2005. IM 2005. 2005 9th IFIP/IEEE International Symposium on. IEEE, 2005, pp. 31–44.

[7] H. Kim, "Efficient and Non-Interactive Hierarchical Key Agreement in WSNs," Int. J. Secur. Its Appl, vol. 7, pp. 159–170, 2013.

[8] S.-W. Lee and H. Kim, "Freshness Consideration of Hierarchical Key Agreement Protocol in WSNs," Int. J. Secur. Its Appl, vol. 8, pp. 81–91, 2014.

[9] R. Gennaro, S. Halevi, H. Krawczyk, T. Rabin, S. Reidt, and S. D. Wolthusen, "Strongly-Resilient and Non-Interactive Hierarchical Key-Agreement in MANETs," in European Symposium on Research in Computer Security. Springer, 2008, pp. 49–65.

[10] 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, no. 2, pp. 611–622, 2013.

[11] A. Shamir, "Identity-based cryptosystems and signature schemes," in Workshop on the theory and application of cryptographic techniques. Springer, 1984, pp. 47–53.

[12] U. M. Maurer and Y. Yacobi, "A non-interactive public-key distribution system," Designs, Codes and Cryptog-raphy, vol. 9, no. 3, pp. 305–316, 1996.

[13] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Annual international cryptol-ogy conference. Springer, 2001, pp. 213–229.

[14] Chaturvedi, A., et al., A New Key Agreement Protocol Using BDP and CSP in Non Commutative Groups. International Journal of Advanced Networking and Applications, 2017. 9(3): p. 3428-3431.