GROUP KEY ESTABLISHMENT FOR SECURE MULTICAST COMMUNICATION FOR THE INTERNET OF THINGS USING SECRET SHARING

[1]AHMED M. ALKUHLANI, [2]S.B. THORAT, [3]JIGNESH BHAYANI, [4]PRITAM AMBEKAR

[1]School of Computational Sciences, S.R.T.M University, Nanded, MS, India
[2]Director of Institute of Technology & Management, Nanded, MS, India
[3,4]Business Intelligence Developer, Canada Goose, Toronto, Ontario, Canada
E-mail: [1]alkohlany1@gmail.com, [2]Suryakant_thorat@yahoo.com, [3]jbhayani@canadagoose.com

Abstract - The Internet of Things or 'IoT' described as a highly interconnected network of heterogeneous devices where all kinds of communications with any type of devices seem to be possible, group communication found to be more efficient in utilizing resources than pairwise communication. However,the imbalance capability of the IoT resource- constrained devices and the heavy cryptographic function of the existing traditional group key establishment protocols promote researchers to develop alternative solutions thatrely on lightweight security primitives to provide end to end security and to protect the information flow in IoT environment. In this paper, we propose a key establishment protocol to secure multicasting in group communication using Slepian-Wolf secret sharing scheme which is an optimized secret sharing technique uses Xor operation instead of polynomials which significantly reduces the storage, communication, and computational costshowever, the security analysis and performance evaluation proves that our protocol resist related group key establishment attacks with a lightweight impact on network and device resources.

Keywords - IoT Secret Sharing, Multicast Security, Key Establishment, IoT Key Management

I. INTRODUCTION

IoT opens new opportunities in several applications domains such as smart home, smart cities, healthcare monitoring, environmental monitoring and smart agriculture [1,2,3,4], an example of smart agriculture is the automatic watering system it is applied in homes, gardens and big farms. Today this is becoming a very important area as more and more areas are facing a shortage of water, hence traditional approaches to watering house plants and gardens are no longer feasible [5, 6].

In certain applications, we require controlling a group of devices rather than a single device, in which the communication and energy costsare reduced.

Multicast communication is an important type of communication for IoT as we mentioned in the automatic watering system example.A group of devices can be controlled by sending one instruction message instead of sending one instruction for each device, specifically, as these devices work in one geographic area having the same type of environmental conditions; therefore, their need for water is the same.Multicast communication reduces the communication and energy cost and achieves efficient utilization of device and network resources [7].

Group key establishment is essential to secure the communication of the multicast group and provide message confidentiality, authenticity,and integrity. IoT relies mostly on sensors which they don't have powerful computational capability, memorycapacity,and also lack consistent energy, however, the security solutions provided for such devices should be lightweight and robust [8].

In this paper we focus on establishing a common secret key between the authorized members of a group, we have used the selpian-wolf secret sharing scheme (SW-SSS) for the group key generation and key reconstruction moreover, through our verification method the scheme maintain a secure and lightweight key delivery with efficient use of resourcesby consuming low storage, communication and computation costs. The rest of the paper is organized as follows review of related IoT key establishment scheme in section 2 in section 3 we discuss the secret sharing concept and present a review of the Slepian-Wolf secret sharing scheme and next in section 4 we present the network model of the smart watering system, in section 5 we present our proposed multicast group key establishment, in section 6 we discuss the security analysis and performance evaluation of the proposed protocol and finally the conclusion is presented in section 7.

II. LITERATURE REVIEW

The Pervasive nature of IoT increases the need for group communication among IoT applications. Compared with pairwise communication, group communication can utilize more efficiently the device and network resources, which is highly required in constraint networks such as IoT.

Some IoT group key establishment protocols are proposed for instance JianShen et al. [9] proposed Key Management Protocol in Body Area Networks, the protocol uses ECC and hash function. All sensors

in each patient considered as a group, the patient controller(pc) controls the key generation and using the shared secret key already computed between the pc and each sensor, the group key is then distributed to all members of the group. The protocol is susceptible to reply attack and due to the use of ECC computational and communication costs is high .in [10] L.Veltri et al. proposed batch-based group key management protocol applied to the Internet of things, the author uses a key distribution center (KDC) to handle the key distribution however, the protocol is intended for highly dynamic environment such as vehicular network where the number of joining/leaving nodes is high. The scheme is time-driven predicts the time of members leaving the group which is consideredimpractical for dynamic environment moreover,the protocol is susceptible to reply attack Also, the author does not specify a mechanism to authenticate the group members [11]. Secret sharing technique is also used for key establishment in IoT for instance Dahshan in [12] proposed a distributed key management scheme for the IoT, the protocol uses secret sharing technique and a digital certificate for key generation and key authentication the author uses certificate authority (CA) to provide each entity with a share of the CA secret key and the corresponding CA public key. The scheme uses ECC which result in high computational, communication and storage costs.

In [13] Porambage et al. proposed Group Key Establishment forIoT Applications based on WSN the author has proposed two protocols in protocol1which was an enhanced version of [14]the author used ECChowever, the author in [15] claim that the Porambage et al protocol is vulnerable to reply attack and node traceability attack.

In our proposed scheme we have used the Slepian-wolf secret sharing scheme (SW-SSS) to overcome the security drawbacks and the extensive computation, storage and communication overhead of the previously related schemes. The SW-SSS construct the group secret key shares by XORing a set of secret blocks which reduces the share size, storage and communication costs moreover, we have used a lightweight verification method based one-way hash function and xor operation to distribute and recover the key secret shares achieving robust mutual authentication key distribution and recovery.

2.1. Security consideration

The main security essentials with group key establishment are to ensure key i) confidentiality, ii) authentication and iii) integrity which in turn ensure a secure communication[16].

One of the essential requirements for group communication is Confidentiality whereas unauthorized members shouldn't be able to recover the key; it must be recovered only by authorized/ legitimate members of the group. Key authentication is to ensure that key is generated from the authorized originator,not from any other unauthorized members or attacker and key integrity is to ensure that the key distributed not altered or changed during communication by any attacker. In the proposed protocol we provide these security essentials as we willprove in the security analysis.

III. NETWORK MODEL

Smart agriculture is one of the application areas that IoT targets, in our network model of the smart watering system, for best utilization of water; the environmental monitoring unit collects data about soil moisture and temperature and delivers aggregated data to the gateway. Based on data received, the gateway can give commands to open/ close water flow to a group of the smart watering devices in a specific area of the farm. Model is presented in figure [1].

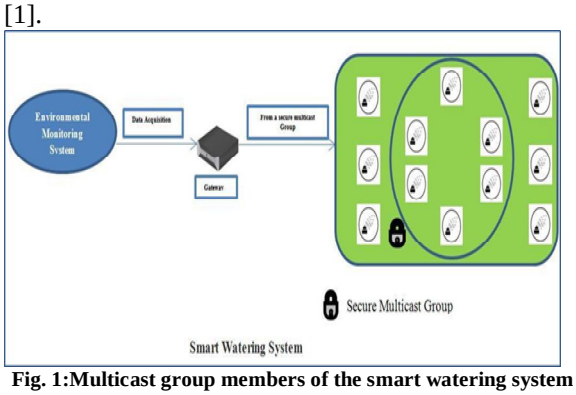


Fig. 1:Multicast group members of the smart watering system

IV. PRELIMINARIES

4.1. Secret Sharing Technique

Secret sharing schemes were introduced by both Blakley [17] and Shamir [18] as a solution for securing cryptographic keys. The concept of secret sharing is to divide the secret intomultiple shares, each share is sent to a different participant. The secret can be reconstructed when certain numbers of shares are combined. For example when a secret S we can divide it inton shares and distribute it to n participants and only qualified sets of participants also called the threshold can recover the key. Traditional Secret Sharing schemes use polynomials for generating/reconstructing the secret S which is considered computationally heavy [16] therefore, not preferred to be used in a constrained environment such IoT. For that, some enhanced secret sharing techniques are proposed using the lightweight X-or operation trying to reduce the memory, computational and communication costs of key generating/reconstructing algorithm.

4.2. Slepian-Wolf Secret Sharing Scheme (SW-SSS)

Initially, Slepian-Wolf coding (SWC) was discovered by David Slepian and Jack Wolf [19] it is a method

used for data compression of theoretically coding two lossless compressed correlated sources. Recently in 2017 based on SWC, Slepian-Wolf Secret Sharing Scheme SW-SSS is proposed [20] as an efficient secret sharing technique in which the computational cost of key generation and secret reconstruction is lightweight also the bit size of each share is small which makes the storage cost at minimum therefore, it is efficient in terms of storage, communication and computation costs which is suitable for constrained network such as the IoT environment.

V. PROPOSED PROTOCOL

In the proposed protocol base on our network model presented in section[3] wherein one-to-many and many-to-one communication is required we propose an efficient multicast group key establishment scheme for IoT the scheme uses Slepian-Wolf secret sharing scheme(SW-SSS) which found efficient in minimizing the storage, communication, and computational costs which can fit for the resource-constrained devices of IoT. The gateway generates the group key and through the SW-SSS key shares are generated then using the pre-shared secrets and the computationally lightweight hash function the group key shares are distributed to the multicast group members.

Our proposed protocol for group key establishment consists of three phases the pre-deployment phase in which the gateway and the other devices of the network are configured the second phase is the key generation and distribution phase and the third phase is the key recovery/reconstruction phase

5.1. Pre-deployment phase

WSN sensors are used within a particular area of interest, before deploying them a certain configuration have to be embedded into them. The field has a powerful device called the gateway and many more of sensors and actuators. In this phase, the system administrator configures the gateway and the network devices in offline mode as follows:-

SA generate a unique identity for each sensor Si ($1<i< b$) in the network and a randomly-generated secret password (key)PWi that will be shared with the gateway.The value b represents the number of sensor nodes within the network that will be deployed in the field. The values Si and PWi will be stored in the sensor tamper-proof memory. Then SA also embeds the previous values (Si, PWi) to the gateway memory.

5.2. Key generation

The gateway is considered a trusted and powerful node in the network it generates random secret key K to secure the multicast group communication. K is divided into equal mblocks to generate n shares be distributed to n sensor members of the group; each sensor will keep a share. As depicted in the algorithm [1] it takes the input of the secret K, the number of shares n and the number of blocks m the size of each block is |b| = K\ m. the algorithm works as follows:

As shown in figure 2 the gateway generates a secret key K=160 bitand divide it equally into m secret blocksK={ $b_0||\ldots\ldots||b_{m-1}$ }theneach share is constructed by XORing three secret blocks,however,the maximum number of Xored blocks b can be determined by in which each block size is |b|=K\m

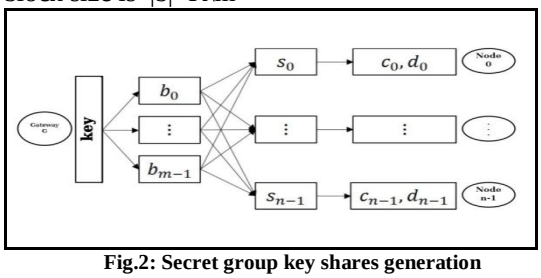


Fig.2: Secret group key shares generation

The idea of selecting the three Xored block for each share is by permuting the index sequence of the proper set (0,….,m- 1) in ascending order and each XORitself is also sorted in an ascending order.

Under our scenario, the multicast group members are 10 wherein =10 which indicates the maximum number of generated shares n, and m refers to the number of secret blocks m=5 and 3 indicates the number of xored secret blocks b. The key size is 160 bit divided into 5 block (0,1,2,3,4) where each block size |b|= 32 bit we need 10 shares for n member in the multicast grouphowever,selecting the three Xored block for each share is by permuting the index sequence of the proper set (0,1,2,3,4) in ascending order then the gateway generate theshares from the following XORs respectively:- (b0⊕b1⊕b2); (b0⊕b1⊕b3); (b0⊕b1⊕b4); (b0⊕b2⊕b3); (b0⊕b2⊕b4); (b0⊕b3⊕b4);(b1⊕b2⊕b3); (b1⊕b2⊕b4); (b1⊕b3⊕b4) (b2⊕b3⊕b4). Key generation algorithm outputspair of a coded block representing the shares ci and it's metadata di for n members where i = {0,....,n-1}. In fact ci represents the index of the share in a list (M) of similar values having the same number of ‘1’bits and the same block size |b|and di represents the number of ‘1’ bits in the share, however, to clarify the idea and for the space constrain we will present an example of block size |b|= 3 bits and d=2 suppose the Xored tuple s=(b0⊕b1⊕b2)= 101 then the list M ={011,101,110}the index of M start from 0 then c is equal to the value 110 which its index is 2 hence the share coded block c=1 and d =2.

After shares of the key are generated the gateway sends the shares (ci, di) to each member of the group using our distribution and verification algorithm.

It should be noted that each node in the group stores only the share index represented by the code block ci and its metadata di, not the key itself which significantly reduces share size and consequently minimizes the storage and communication costs for

the device and the network and maximize the key confidentiality.

Input: m, n, K
Output: $(c_0, d_0), \ldots, (c_{n-1}, d_{n-1})$
1: $K = b_0 \| \ldots \| b_{m-1}$
2: $|b| = |K|/m$
3: $i \leftarrow 0$
4: for $j \leftarrow 0$ to $m-3$ do
5: for $t \leftarrow j+1$ to $m-2$ do
6: for $z \leftarrow t+1$ to $m-1$ do
7: $s_i \leftarrow b_j \oplus b_t \oplus b_z$
8: $d_i \leftarrow$ number of '1' bits in s_i
9: $(c_i, d_i) \leftarrow$ SWC. Encode (s_i, d_i)
10: $i = i+1$
11: if $(i == n-1)$ then
12: return $(c_0, d_0), \ldots, (c_{n-1}, d_{n-1})$
13: end if
14: end for
15: end for
16: end for

Algorithm1.SW-SSS Key generation algorithm

5.3. Distribution and verification algorithm

SW-SSS technique doesn't provide any mechanism for message authentication between the initiator (the gateway) and the group members moreover, the communication medium between the gateway and the group members is unprotected via public wireless channel hence; to prevent any attacker or unauthorized access to the key shares sent to the multicast group members we have to maintain message authentication and message integrity between the gateway and each member of the group.

In the distribution and verification phase the gateway uses one-way hash function and the previously shared password PWi in the setup phase in addition to random numbers R and rj to encrypt the communication between the gateway and each member of the group,however, in this phase mutual authentication between the gateway and the group member is maintained for secure deliveryofthe key shares (ci, di). The steps are shown in the figure [3] and explained in following steps.

After generating the K shares pair (ci, di)$\forall i \in \{1 \ldots n\}$the gateway performs the following:

- Step1. Generate a random number R which will be generated uniquely for each multicast group.
- Step2. Generate a secret group key K.
- Step3.Select a group of sensor nodes by their identity that forms the multicast group, the size of the multicast group is (n), and the list of members in the multicast group are G1= {S1, S2,….Sn}.
- Step4. The gateway generates a fresh time stamp T1 and computes Mri=R ⊕h(Si||PWi|||T1) the value of the random number R is masked with the one-way hash function h(Si||PWi||T1) where Si is the identity of the sensor, PWi is the shared password between the gateway and the sensor Si and T1 is a fresh timestamp.Only the gateway and the sensor Si can compute the hash value and extract the random value R.
- Step5. The gateway computes Mdi=di ⊕h(R||PWi||T1) and Mci=ci ⊕ h(di||PWi||T1) where the secret share pair (ci, di) is masked by the one-way hash function to prevent any type of attacks. We should note that it is computationally infeasible to extract the hashed values by an illegitimate entity. The gateway computes Ei=Mri⊕Mdi and Fi=Mri⊕Mci and finally the Auth=h{ci||di||R||PWi||T1}.
- Step6. The gateway then sends {G1, Ei, Fi, Mri, Auth, T1} to Si where $i \in \{1, \ldots, (n-1)\}$.
- Step7. After receiving the values from the gateway, the Si firstly checks the time |T1 - TC| <ΔT. T1 is the received timestamp from the gateway and TC is the current timestamp of the Si. By computing the |T1 - TC|, the Si checks whether the transmission delay was within the allowed time interval ΔT. If not, the Si aborts any further transmission and sends a rejection message to the gateway.otherwise

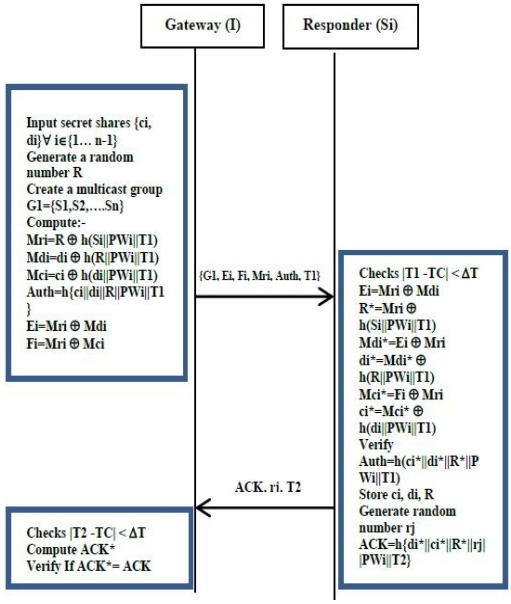


Fig. 3: group key distribution.

- Step8. Each member of the multicast group Si computes R* ci* and di* as explained in figure [3] and from the freshly computed values{R*, ci*,di*, PWi} Si verifies if the received Auth=?h(ci*||di*||R*||PWi||T1) equal to the computed Auth* if so, then the gateway is authenticated and also the shared secret values (ci*,di*) are verified.
- Step9. After each member of the group verifies the authenticity of the gateway and the secret shares (ci, di) it stores R, ci, di and group members identities G1 in the memory then generates a random numberrj, a timestamp T2 and compute ACK=h {di*||ci*||R*||rj||PWi||T2}

and forward the confirmation message {Auth, rj, T2} to the gateway.

- Step10.when the gateway receives {Auth, rj, T2} it first checks the time |T2 - TC| <ΔT. T2 is the received timestamp from the Si and TC is the current timestamp of the gateway. By computing the |T2 - TC|, the gateway checks whether the transmission delay was within the allowed time interval ΔT. If not, the gateway aborts any further transactions and sends a rejection message to the Si. otherwise
- Step11.the gateway compute ACK* and checks If ACK*= ACK then Si is a legitimate member of the group and has the right verified secret share ci and di.

5.1. Key derivation

5.5. Key shares recovery verification

When the secret shares of the key K are generated each member of the group gets a share pair of (ci, di) in contrary, in the secret key K recovery we don't need all shares of n only specific number of the group shares which is called the threshold m are required to derive the key. Therefore, we must ensure that share pairs ci and di are collected from the right legitimate members of the multicast group and prevent any attacker from altering or discovering the secret key shares for that we apply our verification algorithm as follows:-

- Step1. The group member Si computes the messageAuthi= h (R||Si||T1) where the R is the group unique random number generated by the gateway this number is used by the group member to ensure the authenticity of other group members, Si is the sensor identity and T1 is a fresh timestamp generated by Si. Si sends {Request for Key shares (ci, di), Authi, T1}∀ (m-1).
- Step2. Upon the reception of the message each member in (m-1) checks the timestamp interval ΔT to avoid any reply message if ΔT is not within the allowed interval time the message is rejected and Si is considered illegitimate member otherwise, SiVerifies Authi=Auth* if the verification holds then Si computes the masked values of ci and di Ms1=ci⊕ h (Si||R||T2) and Ms2=di⊕ h (Si||R||T2) both ci and diwill be protected by the hash function.
- Step3. Upon the reception of the message {Ms1, Ms2, T2}firstly it will check the time if it is within the allowed time interval Si accept the message otherwise the message is rejected.
- Step4. Si Extract (ci,di) by computing ci = Ms1⊕ h (Si||R||T2) di = Ms2⊕ h (Si||R||T2) ∀Si in (m-1).
- Step5. (ci,di) for every share in m will be fed to SW-SSS reconstruct algorithm to recover the secret key K.

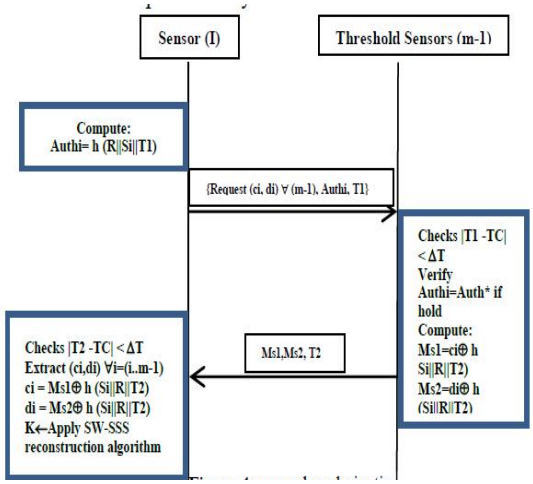


Fig. 4.: group key derivation

5.6. SW-SSS Group key reconstruct algorithm

After getting the key shares from (m-1) members of the group (excluding the initiator of the key retrieval process) to reconstruct the secret K, the Sensor Si executes the Slepian-Wolf Reconstand LocateIndices() algorithms [20]. Reconst algorithmrequiresmgroup membersto participate in the key retrieval process out of n according to our network scenario we require 5 of the group members to provide their share pair (ci, di)to retrieve the key.

VI. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

6.1. Security analysis

In this section, we demonstrate how our proposed scheme withstand against the group key establishment related attacks

6.1.1. key shares mutual Authentication

Mutual authentication is maintained between the key initiator (the gateway) and the multicast group members. The gateway computes the messageAuth=h(ci*||di*||R*||PWi||T1)usingthe shared secretPWi which is a unique secret known only by the gateway and Si and the random numbers R, when Si compute Auth* then it indicates that the message is originated by the gateway no one else therfor, gateway is authenticated.

The gateway in its turn computes the receivedmessage ACK=h{di*||ci*||R*||rj||PWi||T2}if ACK*= ACK then the group member Si is legitimate and mutually authenticated with the gateway. We should note that, since the scheme provides implicit authentication then the man in the middle and impersonation attacks become infeasible.

6.1.2. Key confidentiality

The gateway generated a secret key K then shares are generated according to the multicast group size, the gateway doesn't send the secret share itself, instead, it sends the coded block ci and its metadata di which can be used later by the group member to derive the

group secret key. Moreover, The gateway sends Mdi=di ⊕h(R||PWi) Mci=ci ⊕ h(di||PWi) to each member of the group, suppose an attacker Aintercepted Mdi or Mci from the public channel and tries to extract or guess di or ci , A needs R and PWi to derive ci and di where R is a random number generated for each multicast group and PWi is the password know only to the gateway and the ith member of the group. R and PWi are protected by the one-way hash function which is computationally infeasible to extract ci and di without knowing R and PWi.

Also when reconstructing the key the multicast group share a common random secret key R which in addition to the hash function h() is used to protect the group key confidentiality and prevent any unauthorized attempt to retrieve the key.

6.1.3. Key integrity

Integrity is to ensure the information transmitted is correct and hasn't been modified or alteredby a malicious attacker. The key is protected during communication by the one-way hash function as it is computationally infeasible to modify it without detecting by the authorized members of the group.

6.1.4. Reply attack resilience

Thereplayattack occurs when an attacker intercepts a message and retransmits it again pretending to be a legitimate (gateway or sensor node). In the proposed protocol we use a timestamp in every transmitted message hence our protocol is protected from reply attack.

6.1.5. Gateway/sensor impersonation attack

The gateway sends the message {G1, Ei, Fi, Mri, Auth, T1}to SiSuppose the attacker A intercept the message to impersonate the gateway A needs to know Si,PWi,R and Si,PWi,R,di to compute Ei and Fi respectively also A need to know ci,di,R,PWi to compute Auth all the previous values are unknown to the attacker and protected by the one- way hash function, therefore, A fails to impersonate the gateway.

The same applies to impersonate a sensor node, suppose the attacker intercept {ACK, rj, T2} to compute ACK=h{di*||ci*||R*||rj||PWi||T2} A need to know R,PWi, and ci, di in addition to the strong one-wayfeature of the hash function make it infeasible to compute ACK hence, the protocol is secure against impersonation attack

6.1.6. Forward and backward secrecy

The network model nature of the proposed protocol of the smart watering system is static in which the number of joining/leaving nodes is limited unlike other dynamic networks such as vehicular network where the number of joining/leaving nodes is high.However, in the proposed protocol the group key is updated whenever a leave or joining event occurs so in the case of joining of a new node it will get the new share of the group key and in the case of a node leaving the group which may has a pair share (ci, di) is considered useless because to retrieve the group key we required m pairs of (ci, di) therefore, our scheme provides forward and backward secrecy.

6.2. Performance evaluations with some related secret sharing schemes

Our scheme uses SW-SSS secret sharing technique, for the sake of clarity, we will first present the Performance analysis of the proposed protocol with other related secret sharing schemes the evaluation proves the effectiveness of the proposed protocol. Table [1] summarizes the computational, communication and storage cost comparison of the proposed protocol with the other protocols proposed by other authors.

For the purpose of objective comparison between the three techniques, we considered the key size of 160 bits and network size (n) is 10 sensors and threshold (m) is 5.

Table 1: comparison between SW-SSS and other secret sharing schemes

Protocol		Shamir	X-ORNC	SW-SSS
Computational Cost	Share Generation	O(n log n)⊗	O(mn)(⊕)	O(n)(⊕)
	Shares recovery	$O(m^2)$⊗	$O(m^2)$(⊕)	$O(m^2)$(⊕)
Communication Cost	Share Generation	1600 bit	370 bit	110 bit
	Shares recovery	640 bit	370 bit	44 bit
Storage Cost		160 bit	37 bit	11 bit

6.3. Performance evaluations with the existing IoT Key Establishment Protocols

We have conducted a performance evaluation comparison with some of the existing key establishment protocols in IoT. We present below different computation, communication and storage costs

6.3.1. Computational cost

In our network model for the automatic watering system constrained sensors are involved in the key establishment hence, we have used hash functions and Xor operation to achieve a lightweight computation.The proposed schemeas shown in table [2]outperforms other ECC schemes as the computational cost incurred by our proposed protocol is 11Th +$O(m^2)$⊕, other schemes of P. Porambage et al. [13] computational cost is $10T_h$ + 9 PM + 5 PA

and Dahshan[12] computational cost is $[(n(t+3)+t)\mathcal{M} + ((n(t+3)-1)D]$ both the schemes P. Porambage et al. and dahshan have used ECC elliptic curve which uses heavy mathematical computation functions such as polynomial interpolation, modular arithmetic and pointsmultiplicand and points addition.

Table 2: Computational cost comparison between proposed protocol and other related schemes

Protocol	Computation cost
P. Porambage et al.[13]	$10T_h$ + 9 PM + 5 PA
Dahshan[12]	$[(n(t+3)+t)\mathcal{M} + ((n(t+3)-1)D]$
Proposed	$11T_h + O(m^2)\oplus$

6.3.2. Communication Cost

The communication costs depend on the length of the messages send and received the small the message length the low the communication cost. Communication cost for the gateway is ignored as it is a fully functional and a powerful device;therefore, we maintain a comparison study for the resource-constrained sensors. We considered a key length is 160 bits a hash function is 16 byte, random number is 16 bytes, timestamp is 4 bytes, identity is 2 bytes, password is 16 bytes, prime and ECC points are 16 bytes eachThe total communication cost of messages transmitted and received for key generation, distribution, reconstruction and verification of our protocol for a group of 10 members, is1396byteswherein P. Porambage et al.[13]2004 bytes and Dahshan [12]1472, as a result, our protocol achieve the best optimization in network bandwidth and energy consumption among other schemes as shown in figure [5].

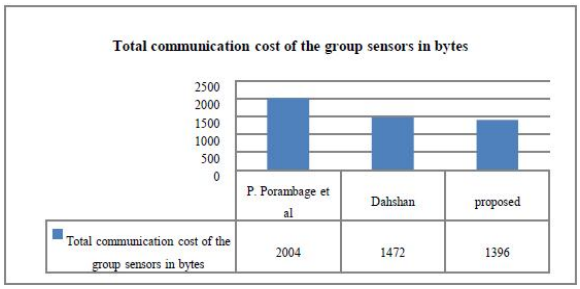


Fig. 5:Total communication costofa group of 10 members

6.3.3. Storage cost

The key share pair(ci, di) represent the index of the sharenot the key itself hence, the size of ci=5 bits and di=6 bits where ci is $\log_2(32)$=5 and di represented by maximum of '1'bits in a share, however, the storage cost is calculated according to the numberof bytes of the values that are stored in a sensor node for a fair and objective comparison we have considered a key length is 160 bits a hash function is 16 byte, random number is 16 bytes, timestamp is 4 bytes, identity is 2 bytes, password is 16 bytes, prime and ECC points are 16 bytes each. The storage overhead of each sensor node in our protocol is 53bytes only which show that our protocol has the minimum storage cost among other protocols as shown in figure [6].

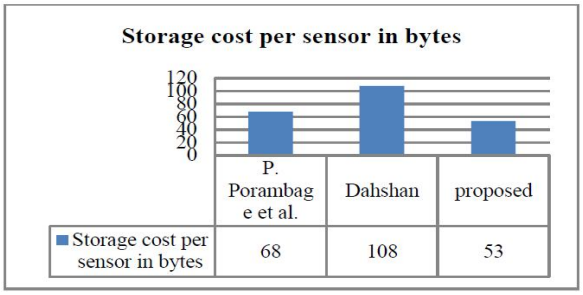


Fig. 6: storage cost per sensor

CONCLUSIONS

In this paper, we proposed Group Key Establishment for multicast communication for the IoT environment we have used the Slepian-wolf secret sharing scheme to generate a secure group key and by the distribution/recovery verification, we guaranteed the key confidentiality, integrity, and authenticity. Unlike other secret sharing schemes the share size ci is optimized as it represents the index of the share which significantly reduces the storage, communication, and computation costs moreover the distribution/recovery verification phase ensure mutual authentication between group members and the gateway.

As a result, the security analysis and the performance evaluation comparison between the proposed scheme and other IoT key establishment schemes shows that our scheme achieves better security and utilize the network and devices resources efficiently.

REFERENCES

[1] Rodrigo Roman et al, "Key management systems for sensor networks in the context of the Internet of Things", Computers & Electrical Engineering · March 2011, DOI: 10.1016/j.compeleceng.2011.01.009

[2] SyeLoongKeoh, "Securing the Internet of Things: A Standardization Perspective", IEEE INTERNET OF THINGS JOURNAL, VOL. 1, NO. 3, JUNE 2014.

[3] RajkumarBuyya and Amir VahidDastjerdi," Internet of Things Principles and Paradigms", 2016 Elsevier Inc.

[4] M. Zhang, T. Yu, G.F. Zhai, Smart transport system based on ''The Internet of Things'', Applied Mechanics and Materials 48–49 (2011) 1073–1076.

[5] Fan TongKe, "Smart Agriculture Based on Cloud Computing and IOT",Journal of Convergence Information Technology(JCIT) Volume 8, Number 2, Jan 2013 Doi: 10.4156/jcit.vol8.issue2.26.

[6] LuisSanchez," SmartSantander: IoT experimentation over a smart city testbed", Computer Networks

[7] Volume 61, 14 March 2014, Pages 217-238, doi.org/10.1016/j.bjp.2013.12.020.

[8] S. Keoh et al," DTLS-Based Multicast Security for Low-Power and Lossy Networks (LLNs)", (Feb. 2014). Available: http://tools.ietf.org/pdf/draft-keohdice-multicast-security-05.

[9] Rodrigo Roman et al, "Key management systems for sensor networks in the context of the Internet of Things",

Computers and Electrical Engineering 37 (2011) 147–159.

[10] J. Shen, M. Sangman,and I. Chung, A Novel Key Management Protocol in Body Area Networks, ICNS 2011: The Seventh International Conference on Networking and Services, pp. 246-251.

[11] L. Veltri, S. Cirani, S. Busanelli and G. Ferrari, A novel batch-based group key management protocol applied to the Internet of Things, Ad Hoc Networks,November 2013, vol. 11, pp. 27242737.

[12] S. Naoui at al.,"Security Analysis of Existing IoT Key Management Protocols", 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA).

[13] H. Dahshan," An elliptic curve Key Management Scheme for Internet of Things", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 20 (2016) pp. 10241-10246.

[14] P. Porambageet al."Group Key Establishment for Enabling Secure Multicast Communication in WSNs", IEEE Access Volume: 3,DIO 10.1109/ACCESS.2015.2474705.

[15] C.-Y. Lee, Z.-H.Wang, L. Harn, and C.-C. Chang, ``Secure key transfer protocol based on secret sharing for group communications," IEICE Trans. Inf. Syst., vol. 94, no. 11, pp. 20692076, 2011.

[16] S. Challa et al." Secure Signature-Based Authenticated Key Establishment Scheme for Future IoT Applications",DOI 10.1109/ACCESS.2017.2676119, IEEE Access

[17] LeinHarn and ChangluLin ,"Authenticated Group Key Transfer Protocol Based on Secret Sharing",IEEE transactions on computers, VOL. 59, NO. 6, JUNE 2010.

[18] G.R. Blakley, "Safeguarding cryptographic keys", AFIPS National Computer Conf., vol. 48, pp. 313-317, 1979.

[19] Shamir, "How to share a secret", Communication of the ACM, vol. 22, no. 11, pp. 612-613, 1979.

[20] D. Slepian, and J. Wolf, "Noiseless coding of correlated information sources", IEEE Trans. Inf. Theory, vol. 19, no. 4, pp. 471-480, 1973.

[21] Tran Phuong Thao et al,"Optimizing Share Size in Efficient and Robust Secret Sharing Scheme for Big Data", journal of IEEE transactions on big data, January 2017.

★★★