Real-time Key Management for Wireless Mesh Network

Le Viet Hoa and Truong Quang Vinh
Ho Chi Minh City University of Technology
tqvinh@hcmut.edu.vn

Abstract — **With the rapid technological development of wireless, wireless mesh network (WMN) is one of the network models which is gradually showing its superiority through several applications and projects thus it is becoming the key of technology for IoT. Due to the vulnerable environment, limited resource and open communication channel, the security design for such networks are significantly challenging. By using real-time synchronization method between transceiver devices in the WMNs, we propose an algorithm based on secret sharing method in which each node generate its key depend on its physical information and the real-time clock. Therefore, we can manage efficiently public and private keys for data encryption and prevent several external attacks to WMNs. We also propose a specific protocol to secure our keys while transferring between devices to prevent internal attacks.**

Index Terms— ***Wireless Mesh Network, key management, wireless encryption, secret sharing, cryptography.***

I. INTRODUCTION

Wireless mesh networks (WMNs) are well-known technology for the Internet of Things (IoT). More and more applications are widely developed based on WMNs systems, such as indoor security system [1], electronic devices management [2], VoIP system [3] and health-care [4]. Composed of low-cost and low-power devices, WMNs involve several issues (e.g., MAC [5], error correction [6] and energy consumption [7]).

They have some characteristics that make them very vulnerable to malicious attacks. Firstly, the wireless channel is open with the same band radio frequency so anyone can participate in communications. The second is that most protocols for WMNs are known publicly such as AODV [8], OLSR [9], GPSR [10] and BATMAN [11]. Therefore, an attacker can easily launch attacks by exploiting security holes in those protocols. A stronger security protocol consumes more power and costs a lot of resource on nodes, which can lead to the bad performance of this network. In most cases, a trade-off must be made between security and performance. Finally, after deploying a WMN, it is difficult to perform continuous surveillance without any fixed infrastructure. Those are the reasons which make WMNs face many various attacks.

There are many security mechanisms have been proposed for such networks in recent years [12]. Most techniques use symmetric key cryptography for security protocol. These techniques require a simple, flexible, and scalable security protocol which has focused on many phases to secure the WMN and to overcome the aforementioned obstacles in WMN. The survey written by Khawla Naji Shnaikat [13] explained three techniques for key management problem in WMN, and classified those techniques upon three phases of a security protocol. Many protocols are designed based on the modeling of those phases and criteria, typically as a protocol proposed by Ramu Kuchipudi called Dynamic key management [14]. These authors researched a dynamic key management method in dynamic WMNs depended on the idea of group key management technique. This technique has also been most concerned by many researchers in the field of mesh network security.

One of the popular key management methods which are classified into group key management technique is secret-sharing schemes proposed by Adi Shamir in 1979 [15]. In this scheme, a master key is a common secret which is known by all the authenticated nodes. This secret is used for many purposes such as cryptography or keyword for access management. Each group of nodes can reconstruct this master key by its private key which is only known by each node. Since Shamir's scheme is simple but effective, it has been improved and used for WMNs in recent years. Lan Yunl et al presented secret sharing-based key management (SSKM) based on Shamir's scheme [16]. This algorithm dynamically generates a different key based on different polynomials from the base station in different periods which can protect the network from the compromised nodes and reduce the high probability of the common keys. Filippo Gandino et al improved Shamir's scheme by adding a new key negotiation routine [17]. This routine is to prevent the case when an adversary compromises a node before the deletion of the master key. Another author proposed a method to combine Shamir's scheme and encryption method using only hash and XOR to reduce the overhead for realistic WMNs which have limited resource [18].

Our proposed method is derived from the idea of building a real-time security method combined with self-organization among the nodes together in WMN. We propose a method to manage real-time keys which is used for AES encryption. Also in this paper, we propose a proper security protocol with the above method, in which the keys can be transmitted in WMN of reliable way. A list of wireless network attacks was implemented to test the sensitive security information of our deployed network.

The remaining sections of this paper are organized as follows: The methods and techniques we used in our proposed algorithm are described in section II. Section III demonstrates how our experiments were implemented, the results of those also are presented in this section. Finally, the conclusion is drawn in Section IV.

II. PROPOSED ALGORITHM

Our security algorithm is designed for the purpose of safely transferring keys and synchronous nodes in WMN. In section 2.1 we will present our key management method based on Adi Shamir's algorithm, the synchronization between nodes by real-time clock helps our keys prevent many external attacks. We also present a protocol used for transferring those keys in WMN; this protocol will focus on preventing the man-in-middle attack and detecting other abnormal activities in this network.

A. Real-time clock key management

The conventional key management methods are easy to be attacked by various attacks such as eavesdropping keys and data, de-authentication, denial-of-service (DoS). Therefore, we propose to use the real-time clock to change continuously private key in the key management of each node and synchronize all nodes in WMN so that these nodes will be completely independent of each other. One of those nodes is the network time protocol (NTP) server and the others are NTP clients. Because of using WMN model, so the NTP data are transferred quickly enough for synchronization. At a certain point, the nodes will together create a unique key and every group of n keys is required to reconstruct the same secret for the encryption and decryption are exactly.

The process of the proposed method is reversely compared with the Adi Shamir's method as shown in Figure 1. In the proposed method, the private key is created first instead of the master key. Therefore, the secret will not be detected when the attacker attacks on any node. Besides that, this secret is constant changes which will make attacks faced many difficulties by using real-time clock modules.

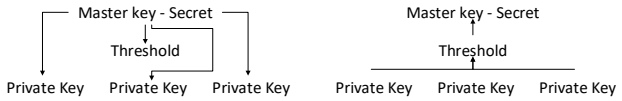

Figure 1: (a) Adi Shamir's secret sharing scheme, (b) Proposed secret sharing scheme

The private key is generated by a unique value depended on each device – MAC address. A threshold level is required for this process. This parameter will be set depending on the number and installation location of nodes in WMN. The process of private key generation is shown in Figure 2.

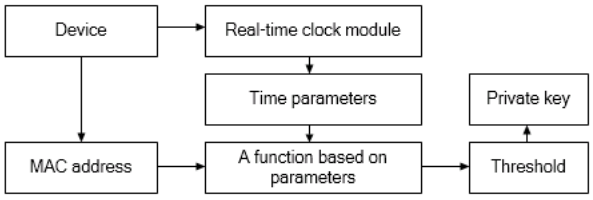

Figure 2: The process of private key generation

The reconstructed secret is implemented after each node has enough keys. It is received from nodes in WMN by simple BATMAN protocol which we will talk in the next section. Lagrange interpolating polynomial was used for our purpose [19]. This is described by the following equations.

For i=1:node

$$S = (S + f(x_i) \prod_{m=0 \wedge m \neq i}^{k-1} (x_m) / (x_m - x_i)) \qquad (1)$$

If we use this original Lagrange interpolating polynomial, there is a security problem: attackers can gain a lot of information about S with every couple key (x_i, $f(x_i)$). They have numbers to guess from instead of an infinite number of natural numbers by using normal basic methods to solve this set of equations.

This problem can be fixed by using finite field arithmetic in a field of size $p \in P : p > S, p > n$. We calculated the couple keys as (x_i, $f(x_i)$ *mod p*) instead of (x_i, $f(x_i)$). The lower one sets p, the lower the number of possible values that the attackers have to guess from to set of S. So we made a small change to our keys generation function and reconstruction function by the following equations:

$$S = (S + p + y.\prod_{k=1 \wedge k \neq i}^{k=node} (-x_k).\delta) \bmod p \qquad (2)$$

$$\delta \times \left(\prod_{k=1 \wedge k \neq i}^{k=node} (x_i - x_k)\right) \bmod p = 1 \qquad (3)$$

where S is the secret value which we need for authentication in WMN. Pair of x, y serves as a key to reconstruct secret as we have presented.

B. Proposed security protocol

The protocol which we use for our key management scheme based on the BATMAN protocol – an efficient protocol used to establish a connection in WMN. Figure 3 describes how our protocol works.

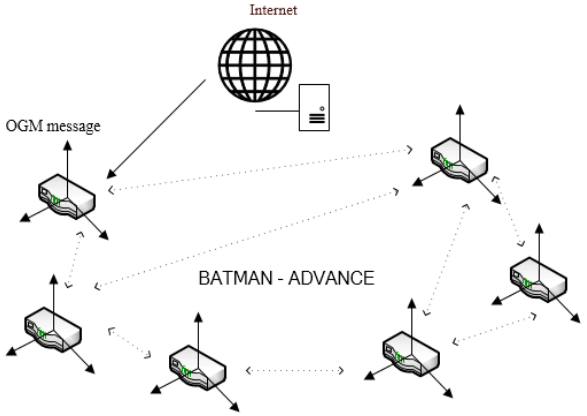

Figure 3: Modified BATMAN protocol

While BATMAN advance protocol work as a communication protocol for sending, receiving data in WMN and detecting node nearby with the same WMN, our proposed protocol use list neighbor nodes of each node in WMN to work for our purpose. Every node sends its private key frequently to neighbor nodes, an authenticated address list has to be created and checked. Those work can be easily done by design a customize package frame on raw debug socket interface (Figure 4).

Source MAC (6bytes)	Destination MAC (6 bytes)	Protocol (2bytes)	Data header (1byte)
Hop count (1byte)	Delay time (8 byte)	Data Length (1byte)	Data

Figure 4: Proposed security protocol header frame

The objectives of proposed security protocol are:

- Encrypt the data by a secret which is reconstructed with keys of nearby neighbor nodes.

- Warn all nodes in the network when there is an intrusion attack in the network.
- Send private key over a man-in-middle node by our frame to increase the range of our protocol.

In order to reach these goals, we combine our protocol and scheme into a multithread program with the flow-graph show as Figure 5 below.

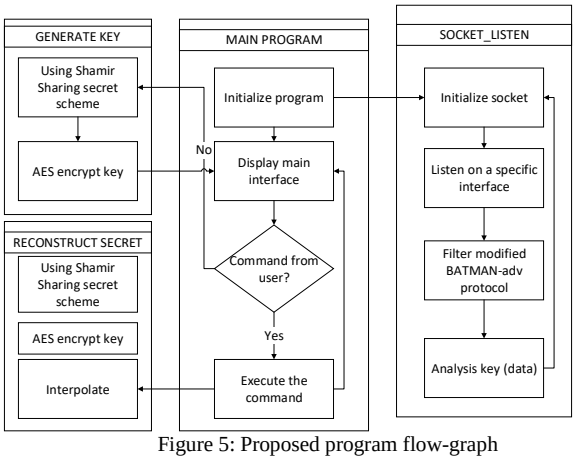

Figure 5: Proposed program flow-graph

Figure 6 and Figure 7 show an example of development with three nodes, node 1 is within the communication range of the others, but the distance between node 0 and node 2 is too long to establish a link.

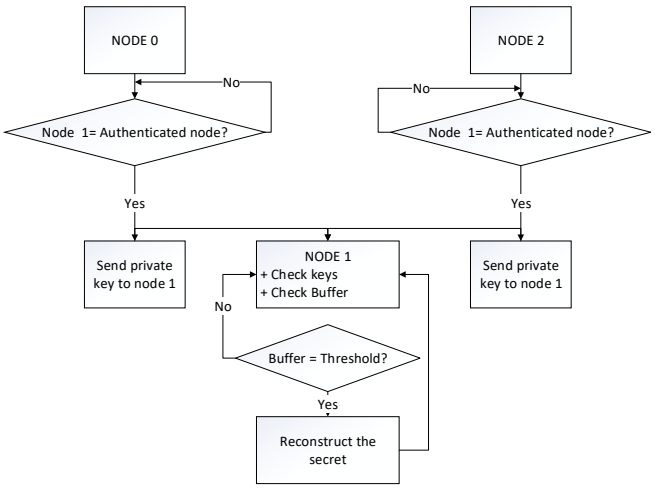

Figure 6: Secret reconstruction of node 1

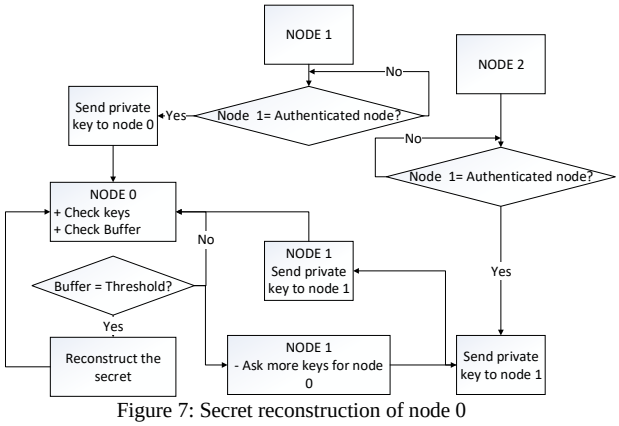

Figure 7: Secret reconstruction of node 0

To prevent the man-in-middle attack as we mentioned before, we encrypt node's keys each time it is transferred to the other node, so there is the problem is that how the requesting node can receive exactly that key with this method. We add a new field "hop count" to our header frame. This field is used for counting the number of nodes which help packet from the requester be transfer to the destination. Then the destination will take that value to encrypt its key <hop count> time s before sending it to the source who requested for more keys. The "key used for encrypting key" is also generated by the real-time method, it must be known by all nodes between source and destination to ensure the decryption on that node is correct. Another issue is that if an intrusion node decrypts more than one time to get the value of the encrypted key, because of this, we set hop count value equal 0 so the intrusion node does not know how many time it has to decrypt for getting the key. Only the owner of the key knows this hop count value.

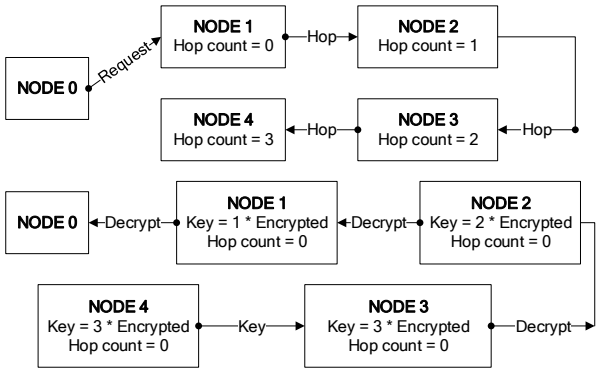

Figure 8: Encryption the key with hop count value

Because of delaying when to transfer the data, it is hard to synchronize the nodes in WMN with our real-time method, the encryption will be incorrect with only NTP system. So we use a buffer at each node in WMN and a field to tell the destination the delay time the request packet sent from the source to destination. When the destination receives the frame, it can send the key it generated at the time the source send out it request packet, the buffer has a responsibility to record all the key value in 1 minute latest. Therefore, if the delay value in WMN is over 1 minute, the requester cannot receive the key from the destination.

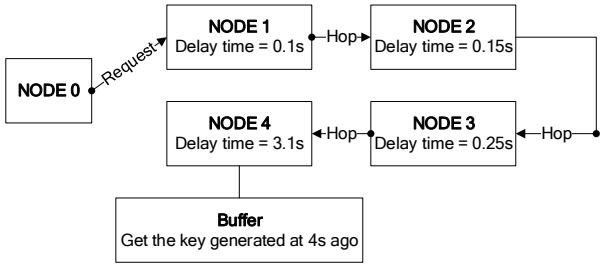

Figure 9: How the buffer and delay time field work

III. EXPERIMENTAL RESULTS

A. System implementation

We consider a WMN consisting of sensor nodes that can work as base stations which will help non-mesh clients communicate together (Figure 10). In this experiment, we bridge wireless local area network interface and mesh network interface together instead of putting WLAN behind the firewall in order to make our work easier. Because we only check if our scheme works perfectly on data link layer, not on the network layer.

Our test-security program has already installed on every sensor nodes, and we put them in the distance. As our scheme, node 1 receives a private key from node 2 and node 3, combine with its key to reconstruct the original secret which is used to encrypting data. This encrypted data is sent to one of the clients of node 4, after decrypting, we compare the decrypted with the original data to see if our scheme work completely. Another parameter which we have to check is secret after reconstruction. We will list all those parameters in the next section.

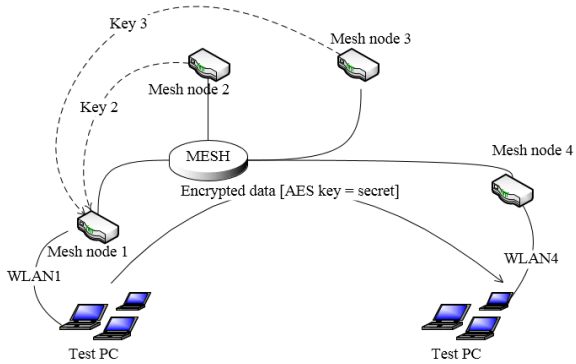

Figure 10: System model

B. Results

1) Secret reconstruction

In this experiment, we set secret – public key of WMN equal exactly with a minute value of UTC time zone. Therefore, the time on every sensor nodes must be set at the same value. Table 1 shows the keys (decrypted keys) collected by node 1 needed for secret reconstruction analysis at the different time. We put a simple function for our experiment secret as follow: Secret = Hour value + Minute value. We executed our program on three nodes in this experiment. This program shows us the value of the generated key of each node, a number of bad nodes this security detected, the time and the secret which was reconstructed at that time. Figure 11 shows all those results of all three nodes at the time of 22:06. Each node has a different private key from the others based on its MAC address. But all of them have the same secret at a certain time.

Table 1
Secret reconstruction analysis

Time	Key 1 (Hex)		Key 2 (Hex)		Key 3 (Hex)		Secret(Dec)
19:45	96	C4	85	F4	26	18	64
19:51	EC	C4	FB	F4	E5	18	70
22:06	8C	C4	6C	F4	D5	18	28

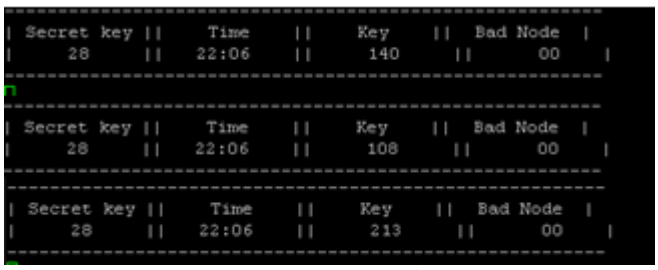

Figure 11: Experimental figure

The secret is reconstructed exactly as the origin with at least 3 private keys of WMN. Therefore, we can run to next step – data encryption, original and decrypted data are the same in both transmitter and receiver if it works correctly, in our case are a client of node 1 and node 3.

2) Data – key encryption analysis

After reconstruction the secret completely, every data which is sent from node 1 will be encrypted by this secret, both encrypted data and decrypted data are shown in table 2. This data is captured at one of the non-mesh clients of wireless local area network node 4.

Table 2
Data encryption analysis

Original text	Secret	Encrypted data	Decrypted data
6bc1bee22e409f9 6e93d7e1173931 72aae2d8a571e0 3ac9c9eb76fac45 af8e5130c81c46a 35ce411e5fbc119 1a0a52eff69f244 5df4f9b17ad2b41 7be66c3710	2aa19fc0 77a19fc 046365b 5bbd288 238	a3e345523cf008 4183a37e6a6c91 fb024b46a8fb64a 774d79bd82a39e b1520cb96204f5 be542e18eae0b4 a809568f9ad5aef 5203c8f30fec0da aef9e31355460	6bc1bee22e409f 96e93d7e11739 3172aae2d8a571 e03ac9c9eb76fa c45af8e5130c81 c46a35ce411e5f bc1191a0a52eff 69f2445df4f9b1 7ad2b417be66c 3710
6bc1bee22e409f9 6e93d7e1173931 72aae2d8a571e0 3ac9c9eb76fac45 af8e5130c81c46a 35ce411e5fbc119 1a0a52eff69f244 5df4f9b17ad2b41 7be66c3710	0021ffc0 7721ffc0 0035350 0666300 62	95e1947945cb24 fc7ca56402a322f 5279569dea7915 53e7c0cc4a9d74 556186ce3688df c8bf5ffe4c61de8 67bd0fc3085430 8fa56dab9a8144 807b82e563ff33	6bc1bee22e409f 96e93d7e11739 3172aae2d8a571 e03ac9c9eb76fa c45af8e5130c81 c46a35ce411e5f bc1191a0a52eff 69f2445df4f9b1 7ad2b417be66c 3710

We had also checked if the keys are secured when they are transmitted in our model (4 nodes with maximum hops equal 2). So then we used an external node which worked as a monitor node to capture the raw package to check if the keys are encrypted. These keys were transmitted from the node had key format “xx:C4” to the node had key format “yy:18”. So then, every time this key hop, its value is encrypted with the key is 18. Table 3 shows the result of this experiment.

Table 3
Key encryption analysis

Original key (hop count = 2)		Encrypted key (AES 128bit)		Decrypted key (2 times)	
		Hop 1 (base 64)	Hop 2 (base 64)		
23	C4	hfr4KW0ps EbZ94pSC Gsn4A==	DFMhDymPlyxoziq6k FpFvQ0/Nxp6p7Ag5i 6H0BadTMA=	23	C4
27	C4	AvoyieF5O iyFv+kqX4 YSbg==	WLLSGdaD/ZuxwLN uUZapgpkauYg2G8o4 8S/eYzriE8o=	27	C4

3) Security analysis

We tested our methods to face types of attacks mainly in general wireless network and in particular wireless mesh network.

The first, let check passive attack - eavesdropping. One thing is sure that if any attack nodes would not have the secret key of this network model is hard to decode data that is collected in the environment, we use ESP8266 kits to collect all the data from our network, the entire data was encrypted. We also tried to collect private keys from authenticated nodes in this network to reconstruct the secret key, but all keys which are transferred in this network had been encrypted with the hop-count parameter we had discussed before. To resolve this issue, an attacker needs to

decrypt those key with the pairs of MAC address and hop-count parameter respectively in this kind of network. Assume that the attackers can decrypt all keys they have collected, there also a major obstacle to them to face is that the keys are constantly changing over time led to the decoding of the troubled him no less.

Secondly, we tested our model with any kinds of active attacks, because of the nature of the connection on the layer 2 of the original BATMAN protocol have already been pretty tight so almost the active attacks up to this model are neutralized, so the impacts of them are only small impacts on single node and easily detected by our protocol when there are abnormal signs from any nodes in our model.

Next, we tried to use the jamming attack to our model. Unfortunately, we have not handled this kind of attack. So in the near future, we will develop our model to overcome this drawback.

Finally, let see how our model handles the man-in-middle attacks. Because we use real-time mainly in our protocol for generating keys, reconstructing the secret and also detect abnormal nodes. So any man-in-middle attacks without synchronized in real time or do not have the ability to interact with the other authenticated nodes in the specified period that we mentioned in the previous section are defined as an abnormal node.

To sum up, the method of attacks can strike on this network model if it knows how the protocol work, however, it requires a process to collect, decrypt, synchronize and analysis accurately complex from the attack nodes.

Table 4 shows a comparison between our algorithm and the others over the security reliability criteria. Our proposed algorithm can prevent many types of attacks which we have discussed before – the other algorithms can not prevent some of them.

Table 4
Security reliability comparison

	Proposed algorithm	Shamir Secret Sharing [15]	ECC [18]
Prevent attacks	- Eavesdropping keys and data. - Deauthentication attack. - DoS attack. - Replay attack. - Man-in-middle attacks.	- Eavesdropping data.	- Eavesdropping keys and data. - Replay attack. - Using MCA handling the fake message.

The original Shamir's algorithm [15] has weakest security reliability in this table because it only prevents attacks focusing on eavesdropping data. The authors of the paper [18] deployed an improved algorithm based on Shamir Secret Sharing. This algorithm prevents not only eavesdropping data but also keys transferred in network models. By using MCA, they stored a hash of shared key with sensor node and adversary cannot get the secret keys. MCA does not have any knowledge of session key and only known to the concerned sensor node. The more resources consumed by the network models if the number of nodes is increased, so the scale of the model deployed by this method is limited. Therefore, our algorithm mainly focuses on extending the scale of the network model with our custom protocol using a minimal buffer on each node that we mentioned in sections above.

IV. CONCLUSION

In this paper, we presented a key management scheme also security protocol for WMNs. In our scheme, we establish secured communication sessions between nodes so they can hide their private keys from the other except the requester. That means not only data also keys were encrypted by combining our scheme with AES encryption. We also use the real time value to change each node's private key constantly. This has caused great difficulty for anyone who wants to find out private keys of WMN. Comparing with existing security protocols and schemes show that our scheme is simple to deployed and it has a better security than.

There remain some problems that should be addressed for this security protocol. We need to reduce the amount of the calculations for the proposed protocol which is deployed on routers with small flash memory. Besides, the WMN structure needs to be improved in order to make the system model work efficiently. Thus, these considerations would be developed in the future work.

REFERENCES

[1] Hsien-Wei Tseng, Yang-Han Lee, Liang-Yu Yen, Su-Yi Yu, and Yi-Lun Chen, "ZigBee (2.4G) Wireless Sensor Network Application on Indoor Intrusion Detection", International Conference on Consumer Electronics-Taiwan, 2015.

[2] Wang Jing, Liu tingting, "Application of wireless sensor network in Yangtze River Basin water environment monitoring", 27th Chinese Control and Decision Conference, 2015.

[3] Manish Tembhurkar and Dr. Latesh Mali, "Energy Efficient V oIP Communication Using WMN Clustering Approach", 8th International Conference on Computer Science & Education, April 26-28, 2013. Colombo, Sri Lanka.

[4] L. Catarinucci, D. de Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, "An iot-aware architecture for smart healthcare systems," IEEE Internet of Things Journal, vol. 2, no. 6, pp. 515–526, Dec 2015.

[5] H. Yan, Y. Zhang, Z. Pang, and L. D. Xu, Super-frame planning and access latency of slotted MAC for industrial WMN in IoT environment," Industrial Informatics, IEEE Transactions on, vol. 10, no. 2, pp. 1242–1251, May 2014.

[6] K. Yu, F. Barac, M. Gidlund, and J. Akerberg, Adaptive forward error correction for best effort wireless sensor networks," in Communications (ICC), 2012 IEEE International Conference on, June 2012, pp. 7104–7109.

[7] U. Kulau, F. Bsching, and L. Wolf, "Undervolting in WMNs: Theory and practice," IEEE Internet of Things Journal, vol. 2, no. 3, pp. 190–198, June 2015.

[8] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF Experimental RFC 3561, July 2003.

[9] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," IETF Experimental RFC 3626, Oct. 2003.

[10] B. N. Karp, H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," ACM/IEEE International Conference on Mobile computing and networking (MobiCom), Aug. 2000.

[11] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better Approach To Mobile Ad-hoc Networking (BATMAN)", IETF Work In Progress sta-Draft, Apr. 2008.

[12] Yun zhou and Yuguang fang, Yanchao Zhang, "Securing wireless sensor networks: a survey", IEEE communication surveys, 3rd Quarter 2008.

[13] Khawla Naji Shnaikat and Ayman Ahmed AlQudah, "Key management techniques in wireless sensor networks", International Journal of Network Security & Its Applications (IJNSA) Vol.6, No.6, November 2014.

[14] Ramu Kuchipudi, Dr. Ahmed Abdul Moiz Qyser and Dr. V .V. S. S. S Balaram, "Latest Developments on Dynamic Key Management for Dynamic Wireless Sensor Networks", International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) – 2016.

[15] Shamir, A. "How to share a secret [J]". Communications of the ACM, 22(11):612-613, 1979.

[16] LAN Yunl, WU Chunying and ZHANG Yiying, "A Secret-sharing-based Key Management in Wireless Sensor Network", 4th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2013.

[17] Filippo Gandino, Renato Ferrero, Bartolomeo Montrucchio and Maurizio Rebaudengo ,"Fast Hierarchical Key Management Scheme with Transitory Master Key for Wireless Sensor Networks", IEEE Internet of Things journal, 2016.

[18] Akansha Singh, Amit K. Awasthi and Karan Singh, "Lightweight Multilevel Key Management Scheme for Large Scale Wireless Sensor Network", InternationalConference on Computing for Sustainable Global Development(INDIACom), 2016

[19] Whittaker, E. T. and Robinson, G. "Lagrange's Formula of Interpolation." The Calculus of Observations: A Treatise on Numerical Mathematics, 4th ed, New York: Dover, pp. 28-30, 1976.