International Journal of Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.02,Issue.17,
November-2013,
Pages:1904-1909

www.semargroups.org,
www.ijsetr.com

Implementation of WSN System Model & Provide the Network Security At Different Attacks using Zero Knowledge Protocol

BASSAM ADNAN ABDULMAHDI[1], DR. WILLIAM JEBERSON[2]

[1]Research Scholar, Dept of IT, Sam Higginbottom Institute of Agriculture, Technology and Science, Allahabad-INDIA,
E-mail: bassamadnan34@gmail.com.

[2]HOD, Dept of CSIT, Sam Higginbottom Institute of Agriculture, Technology and Science, Allahabad-INDIA.

Abstract: Wireless Sensor Networks (WSNs) offer an excellent opportunity to monitor environments, and have a lot of interesting applications, some of which are quite sensitive in nature and require full proof secured environment. The security mechanisms used for wired networks cannot be directly used in sensor networks as there is no user-controlling of each individual node, wireless environment, and more importantly, scarce energy resources. In this paper, we address some of the special security threats and attacks in WSNs. We propose a scheme for detection of distributed sensor cloning attack and use of zero knowledge protocol (ZKP) for verifying the authenticity of the sender sensor nodes. The cloning attack is addressed by attaching a unique fingerprint to each node that depends on the set of neighboring nodes and itself. The fingerprint is attached with every message a sensor node sends. The ZKP is used to ensure non transmission of crucial cryptographic information in the wireless network in order to avoid man-in-the middle (MITM) attack and replay attack. The paper presents a detailed analysis for various scenarios and also analyzes the performance and cryptographic strength.

Keywords: WSN, Cloning Attack, Man-In-The-Middle Attack, Zero Knowledge Protocol.

I. INTRODUCTION

Advances in technology have made it possible to develop sensor nodes which are compact and inexpensive. They are mounted with a variety of sensors and are wireless enabled. Once sensor nodes have been deployed, there will be minimal manual intervention and monitoring. But, when nodes are deployed in a hostile environment and there is no manual monitoring, it creates a security concern. Nodes may be subjected to various physical attacks. The network must be able to autonomously detect, tolerate, and/or avoid these attacks. One important physical attack is the introduction of cloned nodes into the network. When commodity hardware and operating systems are used, it is easy for an adversary to capture legitimate nodes, make clones by copying the cryptographic information, and deploying these clones back into the network. These clones may even be selectively reprogrammed to subvert the network. Individual sensor node contains a light weight processor, cheap hardware components, less memory. Because of these constraints, general-purpose security protocols are hardly appropriate. Public key cryptography is based on RSA approach. The energy consumption and computational latency makes RSA inappropriate for sensor network applications. Security algorithms that are designed specifically for sensor networks are found to be more suitable. The goal of this paper is to develop a security model for wireless sensor networks. We propose a method for identifying the compromised/cloned nodes and also verifying the authenticity of sender sensor nodes in wireless sensor network with the help of zero knowledge protocol.

II. ZERO KNOWLEDGE PROTOCOL

Zero Knowledge Protocols, is an improvement on these situations. The objective is to obtain a system in which it is possible for a prover to convince a verifier of his knowledge of a certain secret without disclosing any information. The present invention relates to Zero Knowledge Protocols that allows the knowledge of some “secret” or private key information in a first party domain to be verified by a second party without imparting the actual secret information or private key to that second party or to any eavesdropping third party. Throughout the present specification, the first party owning the secret information or private key (“s”) and wishing to prove that it has possession of the information will be referred to as the “prover” (“P”); the second party wishing to verify that this is the case without actually receiving knowledge of the secret will be referred to as the “verifier” (“V”). The prover P and verifier V may be any suitable electronic device. The secret information may be any numeric value, hereafter referred to as the secret number of the prover P. ZKP based protocols require less bandwidth, less computational power, and less memory compared to other authentication methods and thus seems to be suitable for WSN.

A. Advantages of Zero Knowledge Protocol

Zero Knowledge Protocols have the following properties:

- The verifier cannot learn anything from the protocol. The verifier does not learn anything in the process of the proof that he could derive from public information by himself. This is the central concept of zero knowledge, i.e., zero amount of knowledge is transferred. There are similar protocols, called Minimum Disclosure Protocols, which relax this property trying to maintain the flow of information to a minimum.
- The prover cannot cheat the verifier. If Pat doesn't know the secret, he can only fool Vani with an incredible amount of luck. The odds that an4 Gerardo I. Simari impostor can cheat the verifier can be made as low as necessary by increasing the number of rounds executed in the protocol.
- The verifier cannot cheat the prover. Vani can't get any information out of the protocol, even if she doesn't stick to the rules. The only thing Vani can do is decide when she accepts that Pat actually knows the secret. The prover will always reveal one solution of many; by doing this he insures that the secret remains intact. This point will become clearer after the presentation of some more complicated systems below.
- The verifier cannot pretend to be the prover to a third party. As stated earlier, no information flows from Pat to Vani. This precludes Vani from trying to masquerade as Pat to a third party. Nevertheless, some ZKP protocols are vulnerable to man-in-the-middle attacks, in which an eavesdropper relays traffic to achieve the desired impersonation effect. A recording of the execution of the protocol is worthless in convincing a third party. Such a recording is identical to a faked one, in which Pat and Vani agreed on the steps before hand.

III. IMPORTANT ATTACKS IN WSN

Though there are various attacks in Wireless Sensor Networks, but certain active attacks that can be detected with our proposed model are as follows:

A. Clone Attack

In clone attack, an adversary may capture a sensor node and copy the cryptographic information to another node known as cloned node. Then this cloned sensor node can be installed to capture the information of the network. The adversary can also inject false information, or manipulate the information passing through cloned nodes. Continuous physical monitoring of nodes is not possible to detect potential tampering and cloning. Thus reliable and fast schemes for detection are necessary to combat these attacks.

B. Man in the Middle Attack

The man-in-the-middle attack (MITM) is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection. The attacker will be able to intercept all messages exchanging between the two victims and inject new ones.

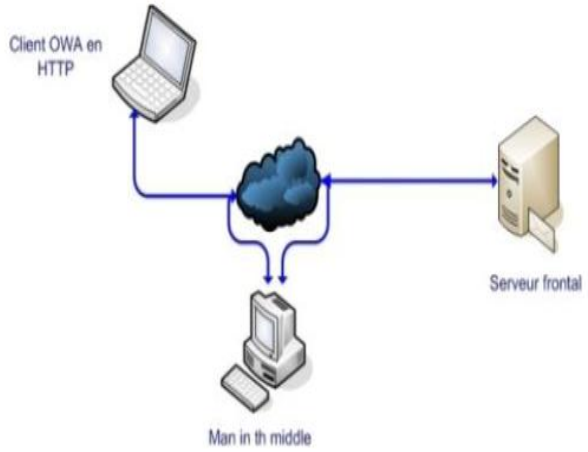

Fig.1. Man in the Middle Attack

C. Replay Attack

A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by adversary who intercepts the data and retransmits it. This type of attack can easily overrule encryption.

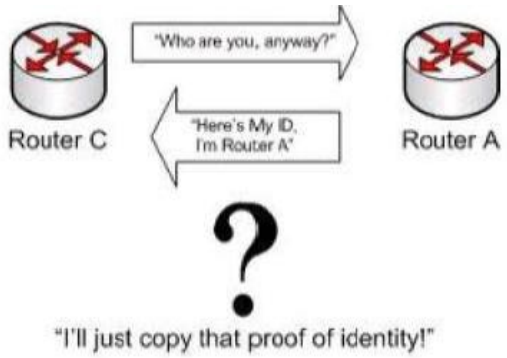

Fig.2. Replay Attack

D. Hello flood Attack

We introduce a novel attack against sensor networks: the HELLO flood. Many protocols require nodes to broadcast HELLO packets to announce themselves to their neighbors, and a node receiving such a packet may assume that it is within (normal) radio range of the sender. This assumption may be false: a laptop-class attacker broadcasting routing or

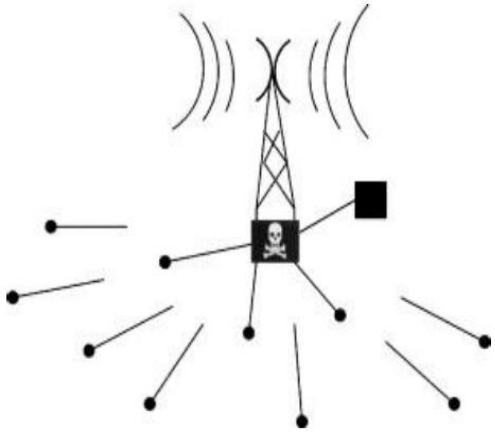

Fig.3. Hello flood Attack

other information with large enough transmission power could convince every node in the network that the adversary is its neighbor.

E. Three-round zero knowledge

The falsity of KEA2 renders vacuous the result of saying that there exists a negligible-error, 3-round ZK argument for WSNs security.

We first consider the protocol of, here called HTP. What has been lost is the proof of soundness (i.e., of negligible error). The simplest thing one could hope for is to re-prove soundness of HTP under KEA3 without modifying the protocol. However, we identify a bug in HTP that renders it unsound. This bug has nothing to do with the assumptions on which the proof of soundness was or can be based. The bug is, however, small and easily fixed. We consider a modified protocol which we call p HTP. We are able to show it is sound (i.e., has negligible error) under KEA3. Since we have modified the protocol we need to re-establish ZK under KEA1 as well, but this is easily done. Arguments we begin by recalling some definitions. An argument for a WSNs L is a two-party protocol in which a polynomial-time prover tries to \convince" a polynomial-time verifier that their common input x belongs to L. In addition to x, the prover has an auxiliary input a. The protocol is a message exchange at the end of which the verifier outputs a bit indicating its decision to accept or reject. The probability (over the coin tosses of both parties) that the verifier accepts is denoted AccP; a V(x). The formal definition follows. A two-party protocol (P; V), where P and V are both polynomial time, is an argument for L with error probability ±: N! [0; 1], if the following conditions are satisfied: Completeness, Soundness and Canonical protocols. The 3-round protocol proposed by, which we call HTP.

IV. PROPOSED MODEL

Nodes are divided into three categories; base station, cluster head and member nodes. Some arbitrary nodes are selected as cluster heads and generation of cluster heads is left to the clustering mechanism (not dealt in this work). Each cluster head knows about its member nodes, while every member node knows its cluster head. Base station stores information of all sensor nodes (including cluster heads). The base station maintains complete topological information about cluster heads and their respective members.

- Base station is powerful enough and cannot be compromised like other nodes of the network.
- There is no communication among the member nodes. Fig.4 describes communications using 3ZKP in the proposed model. The overview of our scheme consists of three main steps categorized into two phases

Base station, cluster head and member nodes are three main nodes in this model. Mostly random nodes are considered as cluster heads. Each and every cluster head had information about its member nodes and vice versa. The information about all sensor nodes which includes cluster heads also is stored in base station. Base station maintains all the topological information about cluster heads and their respective members by communication among member nodes is not possible.

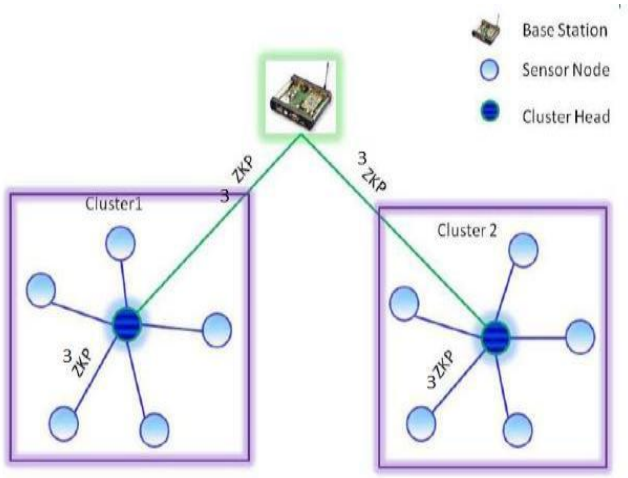

Fig.4. Communications using three-round zero knowledge

A. Pre-deployment phase

For deploying the nodes in the network, we generate a unique fingerprint for each sensor node. I t addressed by combining relative nodes information through a superimposed s-disjunct code and this is preloaded in each node. Due to this each node seems unique from other one. Basically this fingerprint remains secret throughout the process.

B. Post-deployment Phase

A public key N generation by the base station is done after the deployment. Basically this key is used by any two nodes at a given time while communicating. Here base station is third party whereas sender node is prover and receiving node verifier. Each node is assigned a fingerprint which is used as a private key (secret key). Prover and receiver share the public key. Now from base station secret key of the prover from the base station is requested by verifier. The base station will generate a secret code v = s2modN (where s is finger print of the prover and N is the public key). The value of v is given to the verifier on its request. Fingerprint is never shown or transmitted in the network directly during this entire communication process. By using ZKP for k times per communications verifier will continues the authentication process which includes number of verification rounds. Failure of prover for authentication of itself in any one of the k rounds, then it becomes a compromised node. For more effectiveness of protocol it must be passed through large number of rounds. The number s remains private within the domain of the prover. Thus makes it computationally infeasible to derive s from v given v=s2modN.

C. Countermeasures

Outsider attacks and link layer security the majority of outsider attacks against sensor network routing protocols can be prevented by simple link layer encryption and

authentication using a globally shared key. The Sybil attack is no longer relevant because nodes are unwilling to accept even a single identity of the adversary. The majority of selective forwarding and sinkhole attacks are not possible because the adversary is prevented from joining the topology. Link layer acknowledgements can now be authenticated. Major classes of attacks not countered by link layer encryption and authentication mechanisms are wormhole attacks and HELLO flood attacks. Although an adversary is prevented from joining the network, nothing prevents her from using a wormhole to tunnel packets sent by legitimate nodes in one part of the network to legitimate nodes in another part to convince them they are neighbors or by amplifying an overheard broad-cast packet with sufficient power to be received by every node in the network. An insider cannot be prevented from participating in the network, but she should only be able to do so using the identities of the nodes she has compromised. Using a globally shared key allows an insider to masquerade as any (possibly even non-existent) node. Identities must be verified. In the traditional setting, this might be done using public key cryptography, but generating and verifying digital signatures is beyond the capabilities of sensor nodes.

HELLO flood attacks the simplest defense against HELLO flood attacks is to verify the bi-directionality of a link before taking meaningful action based on a message received over that link. However, this countermeasure is less effective when an adversary has a highly sensitive receiver as well as a powerful transmitter. Such an adversary can effectively create a wormhole to every node within range of its transmitter/receiver. Since the links between these nodes and the adversary are bidirectional, the above approach will unlikely being able to locally detect or prevent a HELLO flood? One possible solution to this problem is for every node to authenticate each of its neighbors with an identity verification protocol using a trusted base station. If the protocol sends messages in both directions over the link between the nodes, HELLO floods are prevented when the adversary only has a powerful transmitter because the protocol verifies the bi-directionality of the link. This does not prevent a compromised node with a sensitive receiver and a powerful transmitter from authenticating itself to a large. Authenticated broadcast and flooding since base stations are trustworthy, adversaries must not be able to spoof broadcast or flooded messages from any base station. This requires some level of asymmetry: since every node in the network can potentially be compromised, no node should be able to spoof messages from a base station, yet every node should be able to verify them.

An Authenticated broadcast is also useful for localized node interactions. Many protocols require nodes to broadcast HELLO messages to their neighbors. These messages should be authenticated and impossible to spoof. Proposals for authenticated broadcast intended for use in a more conventional setting either use digital signatures and/or have packet overhead that well exceed the length of typical sensor network packet. LTESLA is a protocol for efficient, authenticated broadcast and flooding that uses only symmetric key cryptography and re-quires minimal packet overhead. LTESLA achieves the asymmetry necessary for authenticated broadcast and flooding by using delayed key disclosure and one-way key chains constructed with a publicly computable cryptographically secure hash function. Replay is prevented because messages authenticated with previously disclosed keys are ignored. LTESLA also requires loose time synchronization.

V. EXPERIMENTAL SETUP

MATLAB has been used to conduct the experiments and verify the proposed model. First, the s-disjunct code matrix, X is generated based on the number of nodes (which is always more than the number of nodes to be deployed in the network). Each column in the matrix corresponds to codeword of each node. Next, a data structure is generated and maintained by the base station corresponding to every sensor node, and their fingerprints. If the outcome of verification is true then the prover is authenticated and later verified for k times to validate it, otherwise the base station is alerted about the compromised prover node, which is later isolated from the network.

A. Security Analysis of the Proposed Model

1. Cloning Attack

Case 1: When the cloned node uses any other existing id with same finger print when a node is compromised and cloned, its clones are launched in the network and try to take part in the communication. The cloned nodes will not be

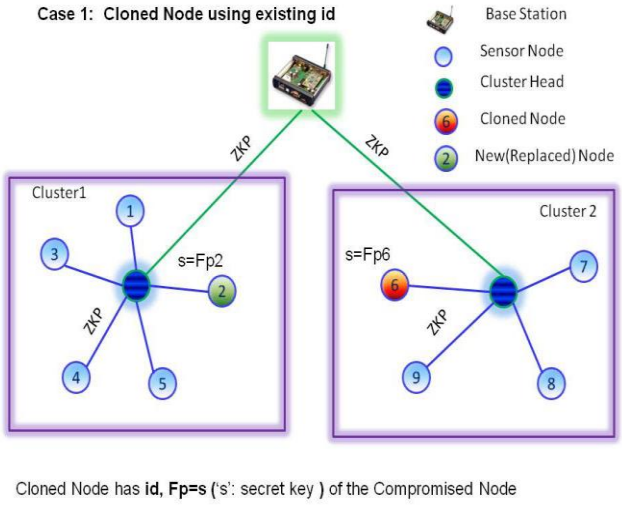

Fig.5. Case1 of security analysis

able to communicate with any other node until and unless it is verified (by cluster head if it is a cloned member node and base station if it is a cloned cluster head). This scenario is explained in Fig.6. In Fig. 5, node '6' of cluster '2' is cloned and placed in cluster '1' with a new id '2'. Since the cloned node uses the finger prints' of node '6', it will fail to authenticate itself during communication through ZKP.

Case 2: When the cloned node uses same id with same finger print If it uses the same id '6', the cluster head of

cluster 1 will reject any communication as node '6' as it is not a member of cluster '1'.The base station which will detect immediately at the initiation of the communication request. This scenario is depicted in Fig.6.

Case 3: When cloned node uses existing id with a different finger print The cloned node having some existing Id can always be detected by the neighboring nodes (cluster heads) as the secret finger print of the cloned node will not match with the finger print possessed by the neighbors.

Case 4: When a cloned node behaves as a cluster head the cluster heads communicate with base station which has all information about the nodes. The base station becomes the verifier and poses the challenge question to the cloned cluster head and detects the cloning attack through ZKP.

2. Man-In-The-Middle Attack

In this type of attack, even though the attacker tries to make independent connections with the victims, it will not be able to authenticate itself to the end nodes (prover and verifier) since it has no clue of the fingerprint of the two end nodes. In our model, the finger print of a node never gets transmitted and the intruder never gets a chance to know them. Even if the attacker tries to generate a finger

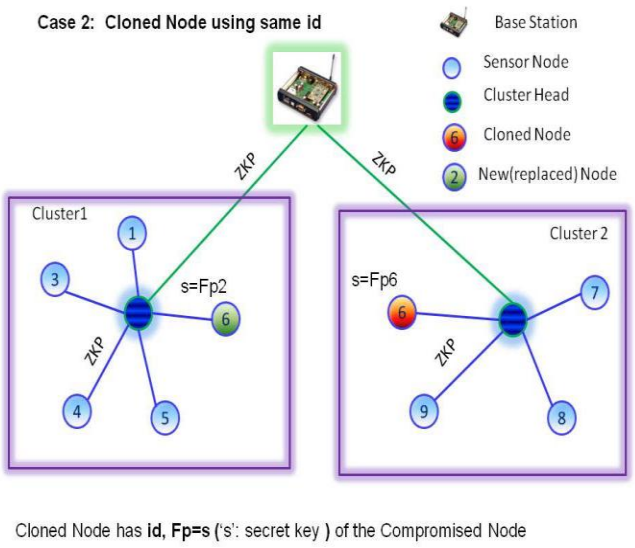

Fig.6. Case2 of security analysis

print in some brute force method, it will not be able to escape the check as every time a new public key N and a new random challenge question will be used.

3. Replay Attack

In this attack, an intruder tries to replay the earlier communication and authenticate it to the verifier. But, as the verifier will be sending different challenge values for each communication, replaying earlier communication will not authenticate the sender.

B. Performance Analysis

The fingerprint generation requires only O(n) computations as simple binary operations are involved in the local FP computation. It has extremely low computation overhead. ZKP also has lighter computational requirement than public key protocols (much faster than RSA). Unlike earlier schemes, the message length in the proposed model is also less as it does not send the finger print with every message. But, in our proposed model, the number of communications increases as it need to communicate with base station to obtain the function of the finger print of the prover to authenticate.

C. Cryptographic Strength

The cryptographic strength of ZKP is based on few hard to solve problems; the one which we have used in our scheme is based on the problem of factoring large numbers that are product of two or more large (hundreds of bits) primes. The values of the public key also changes with every communication, making it more difficult for the attacker to guess it. The prover also generates a random number and the challenge also changes randomly. Thus, with a changed public key, challenge question from verifier and a new random number from the prover, it becomes extremely difficult for the attacker to break the security.

VI. CONCLUSIONS

In this project, we proposed a new security model to address three important active attacks namely cloning attack, MITM attack and Replay attack. We used the concept of zero Knowledge protocol which ensures non-transmission of crucial information between the prover and verifier. The proposed model uses social finger print based on s-disjunct code together with ZKP to detect clone attacks and avoid MITM and replay attack. We analyzed various attack scenarios, cryptographic strength and performance of the proposed model. In future, we propose to extend our work to detect the passive attacks also and evaluate performance in real time using Tiny OS.

VII. REFERENCES

[1] Kai Xing Fang, Liu Xiuzhen, Cheng David, H. C. Du, Real- Time Detection of Clone Attacks in Wireless Sensor Networks, Proceedings of the 28th International Conference on Distributed Computing Systems, 2008, Pages 3-10.

[2] Nikos Komninos, Dimitris Vergados, Christos Douligeris, Detecting Unauthorized and Compromised Nodes in Mobile Adhoc Networks Journal of Ad Hoc Networks, Volume 5, Issue 3, April 2007, Pages: 289-298.

[3] Klempous Ryszard, Nikodem Jan, Radosz Lukasz, Raus Norbert, Adaptive Misbehavior Detection in Wireless Sensors Network Based on Local Community Agreement, 14th Annual IEEE International Conference and Workshops on the Engineering of Computer- Based systems, ECBS'2007, 2007, Page(s):153-160.

[4] Krontiris Ioannis, Tassos Dimitriou and Felix C. Freiling, Towards Intrusion detection In Wireless Sensor Networks, In Proc. of the 13th European Wireless Conference, 2007.

[5] Joseph Binder, Hans Peter Bischof, Zero Knowledge Proofs of Identity for Ad Hoc Wireless Networks An In-Depth Study, Technical Report, 2003. http://www.cs.rit.edu/ jsb7384/zkp-survey.pdf.

[6] A. A. Taleb, Dhiraj K. Pradhan and T. Kocak A Technique to Identify and Substitute Faulty Nodes in Wireless Sensor Networks Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications, 2009, Pages: 346-351.

[7] Klempous R.; Nikodem J.; Radosz, L.; Raus, N. Byzantine Algorithms in Wireless Sensors Network, Wroclaw Univ. of Technol., Wroclaw; Information and Automation, 2006. ICIA 2006.International Conference on, 15-17 Dec. 2006, pages: 319-324.

[8] I. Krontiris, Z. Benenson, T. Giannetsos, F. C. Freiling, and T. Dimitriou, Cooperative Intrusion Detection in Wireless Sensor Networks, in Proc. EWSN'09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 263-278.

[9] A. G. Dyachkov and V. V. Rykov., Optimal superimposed codes and designs for Renyis Search Model. Journal of Statistical Planning and Inference, 100(2):281-302, 2002.

[10] A. J. Macula., A simple construction of d-disjunct matrices with certain constant weights Discrete Math., 162(13):311-312, 1996.

[11] K. Xing, X. Cheng, L. Ma, and Q. Liang, Superimposed Code Based Channel Assignment in Multi-radio Multi-channel Wireless Mesh Networks. In MobiCom'07, pages 15-26, 2007.

[12] Md. Moniruzzaman, Md. Junaid Arafeen, Saugata Bose, Overview of Wireless Sensor Networks: Detection of Cloned Node Using RM, LSN, SET, Bloom filter and AICN Protocol and Comparing.