Wireless Sensor Network Security model using Zero Knowledge Protocol

Tejaswini.C.M
Department of ISE
City engineering college,Bangalore
vishveshvaraya Technological
University, Karnataka,India
tejaswinigowda.c.m@gmail.com

Chaitra.S
Department of CSE
City engineering college,Bangalore
vishveshvaraya technological
University,karnataka,India
chaitra.sgowda.pdk@gmail.com

***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 [1],[2],[3]. 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 [5],[15].

II. PRELIMINARIES

A. Superimposed s-disjunct code

In this section, we introduce the basics of superimposed s-disjunct code, which incorporates social characterstics and used to generate fingerprint for each sensor node [1]. These fingerprints are subsequently used to detect clone attack. Let **X** be a m X n binary matrix. In this paper, we consider a matrix **X** with a constant column weight ω and a constant row weight λ. Then,

$$\sum_{i=1}^{m} x_{i,j} = \omega$$

$$\sum_{j=1}^{n} x_{i,j} = \lambda$$

Where $1 \leq i \leq m, 1 \leq j \leq n$. The binary matrix **X** can be used to define a binary codeword, with each column $\mathbf{X}_j = (x_{1,j}, x_{2,j}, \ldots, x_{m,j})^T$.

Definition 1 Given two binary codewords **y** $= (y_1, y_2, \cdots, y_m)^T$ and **z** $= (z_1, z_2, \ldots, z_m)^T$, we say that **y** covers **z** if the boolean sum (logic OR operation) of **y** and **z** equals **y**, i.e. **y** $\vee$ **z**=**y**.

Definition 2 An mXn binary matrix X defines a superimposed code of length m, size n, strength s (1<s<m), and list size L $(1 \leq L \leq m - s)$, if the boolean sum of any s-subset of columns of **X** can cover no more than L columns of **X** which are not in the s-subset. This code is also called as *(s,L,m)*-code of size n.

Definition 3 A binary matrix **X** defines an s-disjunct code if and only if the boolean sum of any s-subset of columns

of X does not cover any other column of X that are not in the s-subset.
According to the s-disjunct characteristic of superimposed s-disjunct codes, the following important property, can be employed to compute fingerprints to detect clone attacks.
Property 1 Given a superimposed s-disjunct code **X**, for any s -subset of columns of **X**, there exists at least one row in **X** that intersects all the s columns with a value 0.
Generation of a good superimposed s-disjunct code has been extensively studied in literature ([9, 10, 11, 13]). We use a superimposed s-disjunct code with constant weight in our model.

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:

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 is necessary to combat these attacks [1],[13].

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.

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.

IV. ZERO KNOWLEDGE PROTOCOL

Zero-knowledge protocol allow identification, key exchange and other basic cryptographic operations to be implemented without revealing any secret information during the conversation and with smaller computational requirements in comparison to public key protocols. Thus ZKP seems to be very attractive for resource constrained devices. ZKP allows one party to prove its knowledge of a secret to another party without ever revealing the secret. ZKP is an interactive proof system which involves a prover, P and verifier, V. The role of the prover is to convince the verifier of some secret through a series of communications. Each communication involves a challenge, or question, from the verifier and a response, or answer, from the prover. 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 [5].

Basic Mechanism of Zero Knowledge Protocol

The use and implementaion of ZKP in systems and devices that have restricted computational resource are described in [15].
The prover P and the verifier V may use some numeric value, referred as the secret number of the prover P. Conventionally, the prover will offer a computational intensive mathematical problem, and the verifier will ask for one of the many possible solutions to the problem. If the prover knows critical information relating to the solution, it provides any one of the requested available solutions on demand. If the prover does not know the critical information, it is computationally infeasible for it to always provide the requested solution to the verifier. Usually, ZKP rely on some hard mathematical problems such as the factorisation of integers or the discrete logarithm problem.

V. PROPOSED MODEL

Assumptions

- 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 [1]. • There is no communication among the member nodes. Figure 1 describes communications using ZKP in the proposed model.

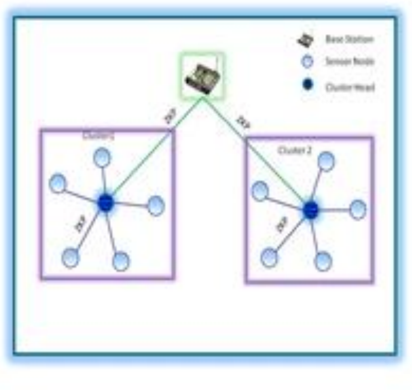


Fig. 1. Communications in the proposed model

The overview of our scheme consists of three main steps categorized into two phases

Pre-deployment Phase

Prior to deployment of the nodes in the network, a unique fingerprint for each sensor node is computed by incorporating the neighborhood information through a superimposed s-disjunct code [9],[10] and is preloaded in each node. The fingerprint allows each node to be different from others and this fingerprint will remain a secret and acts as the private key for the sensor node throughout the communication process.

Generation of unique fingerprint for each node

The base station is assumed to be aware of the topology of the network and all neighborhood information. Before deployment, the base station computes the finger print for each node in the network. For every node u, base station finds its neighborhood information. In our approach, the neighbourhood Ngh(u) should satisfy ng<*s*, where ng is the number of sensor nodes in Ngh(u), *s* is the strength of the superimposed code X. Finger print for sensor node u is computed by considering the code words of all node v which are in the Ngh(u). Given a sensor node u, base station computes u's fingerprint as follows. Let $X^u = X^1, X^2, ..., X^{ng}$ denotes the codeword set of the nodes in Ngh(u), where i denotes the codeword of u's *i*-th closest neighbour. Out of all X^u, the boolean sum of s-closest neighbors of s),is computed first. According to the property of the superimposed s-disjunct code, the resulting vector should contain at least one element with a value 0. These zero elements imply the relationship among the s neighbors, which represent the social characteristic of sensor node u. Motivated by this observation, we use binary representation of the position of a zero element in the boolean sum of

s as the social fingerprint of u. Intuitively, the social fingerprint should be stronger if more information from Ngh(u) is brought in during the fingerprint computation [1]. Base station repeats this procedure mentioned in figure 2 to compute the fingerprint for each node u in the network [1]. The method starts with a s-subset of X(u) that contains

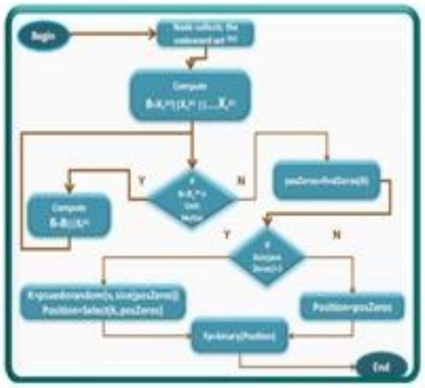
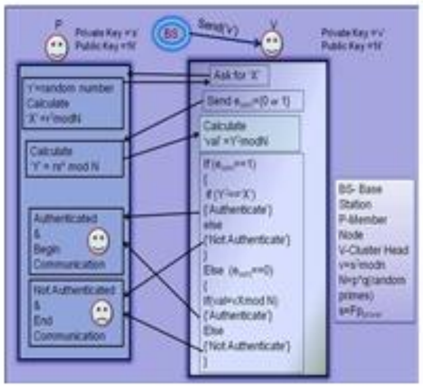


Fig. 2. Method for Generation of fingerprint

the code words of the s closest neighbours of sensor node u, and expands the subset until any further increment will not have a zero element in the boolean sum. For the subset resulting from the last increment, boolean sum is computed and position of one of the zero elements in the resulting sum is selected. The binary equivalent of the position value is denoted as the finger print of node u. By taking u's Id as seed for the pseudo random function, base station is able to compute unique positions for zero element [1].

Post-deployment Phase

After deployment, a public key N (which is a multiplication of large prime numbers) is generated by the base station which will be shared among any two nodes that will be communicating at a given time. During the communication the sender node acts as the prover while the receiver node acts as the verifier. The base station acts as the trusted third party. Each node is assigned a fingerprint which is used as a private key (secret key). The public key N is shared among the sender (prover) and the receiver (verifier). Verifier will request for the secret key of the prover from the base station. The base station will generate a secret code $v = s^2 \bmod N$ (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. During the entire communication process the secret i.e. fingerprint is never revealed or transmitted in the network directly. As explained, in the earlier section, the entire process of authentication is carried out between the prover and the verifier until the receiver node is sure about the authenticity of the sender node. The verifier will continue the process of authentication involving a series of verification rounds using ZKP for k times/communications. The value of k depends on the verifier. If the prover fails to authenticate itself in any one of the k rounds, then it is considered to be a compromised node. This scheme will be very helpful in dealing with the cloning attacks [6],[7],[8].



P
Private Key ='s'
Public Key ='N'
BS
Send('v')
V
Private Key ='v'
Public Key ='N'
'Y'=random number Calculate 'X' =r²modN
Calculate 'Y' = re^e mod N
Authenticated & Begin Communication
Not Authenticated & End Communication
Ask for 'X'
Send e = (0 or 1)
Calculate 'val' =Y²modN
If (e==1) { if ('Y'=='X') ('Authenticate') else ('Not Authenticate') } Else (e==0) { If(val==vXmod N) ('Authenticate') Else ('Not Authenticate') }
BS- Base Station
P-Member Node
V-Cluster Head
v=s²modn
N=p*q(random primes)
s=Fp

Fig. 3. Implementation of ZKP in our Proposed Scheme

To be effective, the protocol is conventionally carried out over a reasonably large number of rounds (or trials or communications). Each round gives V an increasing degree of confidence that P knows the correct number *s*. The number *s* remains private within the domain of the prover. Since N is a product of at least two large primes unknown to V (typically of 1024 or 2048 bit number), it is extremely difficult to factorise, and thus makes it computationally infeasible to derive s from v given $v = s^2 \bmod N$.

1) **Stage 1**: The prover P chooses a random number r, calculates $r^2 \bmod N$ and transmits to the verifier V. 2) **Stage 2**: The verifier V now chooses one of two questions to ask the prover P. The verifier V can ask either for the value of the product (rs) mod N, or for the value of r that the prover has just chosen. This is generally performed by V, sending a bit e to P, indicating its choice of question, refered to as the challenge, such that the prover P has to provide the answer, $y = rs^e \bmod N$, where $e \in (0,1)$. P can answer both correctly if it knows the secret s.

3) **Stage 3**: The prover P provides $y = rs^e \bmod N$ as requested and the verifier checks the result as follows. If the challenge is for e=1, the verifier expects to have recieved rs mod N. The verifier cannot deduce any information about s from this, because r is a random number not known to V. Therefore, the verifier checks $y^2 \bmod N$, which should be $((rs \bmod N)^2 \bmod N)$ is the same as $r^2 * s^2 \bmod N$. The verifier received r^2 from P in stage 1 of this round, and gets *v* from the trusted third party.
If the challenge is for $e = 0$, the verifier expects to have received r, and checks that its square matches the value of r mod N provided in stage 1.

All the above three stages are discussed in Fig.3.

The complete protocol requires execution of a sufficient number of rounds to satisfy V that it is communicating with P, and not an impersonator. Given that the impersonator has a 50:50 chance of selecting the correct strategy in each round. If the protocol requires 10 rounds, i.e. 10 sequential correct responses to the challenges e=(0,1), the odds of an impersonator that does not know s successfully proving to V is less than 2^{-10}.
Each round requires the use of a new value of randon number r. The protocol also requires that the response to a challenge be provided within a time limit such that it becomes computationally infeasible for an impersonator to answer to the challenge by using some brute force method.

VI. 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.

VII. Security analysis of the Proposed Model

CLONING ATTACK

Case 1:When the cloned node uses any other exisiting 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 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 Figure 5. In Figure 4, node '6' of cluster '2' is cloned and placed in cluster '1' with a new id '2'. Since the cloned node uses the finger print 's' 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 Figure 5.

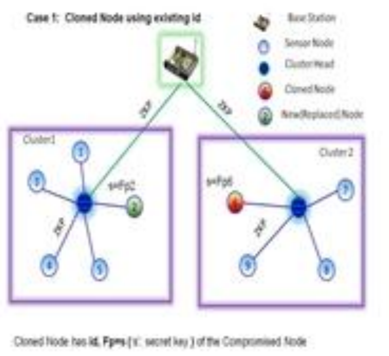


Fig. 4. Case1 of security analysis

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 quesion to the cloned cluster head and detects the cloning attack through ZKP.

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

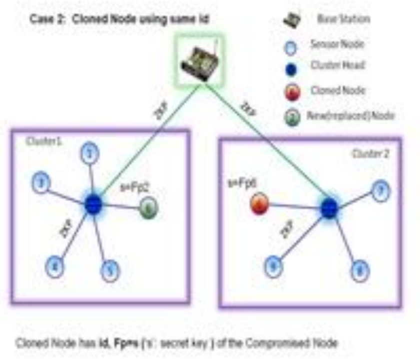


Fig. 5. Case2 of security analysis

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 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.

REPLAY ATTACK

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

Performance Analysis:

The fingerprint generation [1] 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.

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.

VIII. CONCLUSIONS

In this paper, 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 knowlege 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 analysed 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 TinyOS and Tossim.

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*

[13] H.Choi, S.Zhu, and T.Laporta.,*Set: Detecting Node Clones in Sensor Networks*. InSecureComm'07, 2007.

[14] Goldreich, O., Micali, S., and Wigderson, *Proofs That Yield Nothing But Their Validity Or All Languages in NP Have Zero-knowledge Proof Systems*, Journal of the ACM, Vol. 38, No. 1, pp.691-729, 1991.

[15] Tuyls, Pim T. (Mol, BE), Murray, Bruce (Eastleigh, GB),*Efficient Implementation of Zero Knowledge Protocols* ,United States NXP B.V. (Eindhoven, NL) 7555646 ,June 2009 ,http://www.freepatentsonline.com/7555646.html