Tracking of Non cooperative Target to Prevent the Gang Injection of False Data Attacks in Wireless Sensor Networks

Geethu K Mohan & Sreekantha Kumar V.P

Department of Computer Science and Engineering, KCG College Of Technology, Karapakkam, Chennai
E-mail : geethuunni@gmail.com, vpsreekanth@yahoo.com

***Abstract* – Wireless sensor network (WSN) is a collection of heterogeneous sensor nodes having limited computation capacity, restricted memory space, limited power, and short range of communication which all are connected by wireless network. It is vulnerable to security attacks, attacks may be in the form of injection of false data, injection of gang of false data, DOS attack, and selective forwarding etc. attacks are via only the compromised nodes. The integrity of node is broken by forcefully transferring the bogus information through compromised nodes. This paper addresses the gang injection of false data and measures to mitigate it in the Wireless Sensor Network so that the efficiency of the whole network can be maintained without drastic change. The scheme which is demonstrated here is to achieve not only high en-route filtering probability but also high reliability with multi-reports and high scalability with heterogeneous sensors. This scheme can reduce the processing overhead in each node and also in the sink. The primary goal of the paper is to mitigate the gang injection of faulty data attack from mobile compromised nodes. The secondary goal of the paper deals with the identification of compromised node.**

***Key Words - Wireless Sensor Network, Network Security, False Data Injection, Compromised Nodes, Energy Conservation*.**

I. INTRODUCTION

A Wireless Sensor Network (WSN) is network of small devices, called sensor nodes, that are embedded in the real world for observations related to an application and are connected by a wireless network. It have a number of application in medical- health care, disaster management, military service, weather forecasting, environmental applications, tracking of non- cooperative targets etc.

II. EXISTING SYSTEM

Wireless Sensor Networks are vulnerable to a variety of attacks like altered routing information, selective forwarding, sink hole, worm hole, false data injection etc. Attacks are classified mainly into two categories, Routing attacks and attacks on transit.

A. Problem Definition

False data attacks in a Wireless Sensor Network are through compromised nodes. Compromised nodes are nodes those have lost their identity. Any intruder can easily pass false data through this nodes thereby they can drop the equilibrium of the sensor network. Each node in a Wireless Sensor Network is deployed by loading their unique identity verification code and a number of parameters that together identify the object uniquely in the network. When a node compromised, it is easier for the intruder to inject false messages, which in turn results false data interpretation and it produces false result. Sensed data is transmitted through the network and routed by router, while forwarding each packet through the network it consumes considerable amount of energy as well as time. Considering the sensor node, each node can hold a limited amount of energy and data forwarding consumes a part of the stored energy. If it is a false data it is a loss of energy as well as time.

B. Preventing Gang Injection of False Data in Wireless Sensor Network

From figure 1, the node which passes the sensed data can be an active node or it can be a compromised node. Gang injection of false data can be interpreted like the following,

1. A compromised node is going to inject a false report/ data.

2. For such an injection all the neighboring compromised nodes will group together and they all will inject the false report.
3. If only one node is injecting the false report, it will be easy to detect the false report.

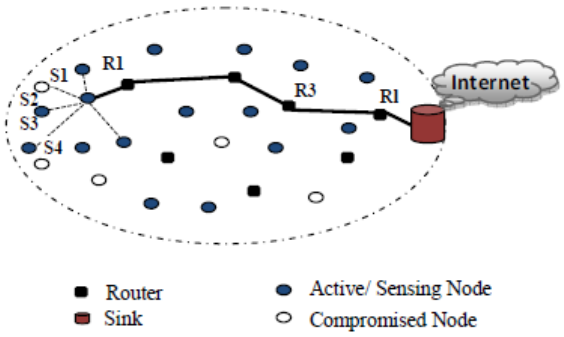

Fig 1 Sample Wireless Sensor Network

III. PROPOSED SYSTEM

It will be better if identify the false data as early as possible. It will reduce the effort in the intermediate nodes at the same time sink. A number of filtering mechanisms are available today. But most of the mechanisms uses symmetric key, hence key destruction is easier. Proposed system is an effective method for filter 90% of bogus data in a Wireless Sensor Network. Still, gang injection of false data cannot be prevented effectively because of the mobility of the sensor nodes.

The paper explains how can resist gang injection of false data. To prevent Gang injection of false data*[4]* it's necessary to identify the compromised sensor nodes. Since all the nodes poses mobility, it will be too difficult to track such nodes.

The second half of the paper explains about the compromised node detection. The compromised node detection can be done via many ways. The survey for compromised node detection indicates two main methods, which are listed below.

1. Hardware Based Attestation Method.
2. Software based Attestation Method.

Remote software attestation method is considered in this paper for further growth of this project.

The set of sensor nodes S = {S0, S1, S2…… Sk} which is a collection of active / sensing nodes and compromised nodes. Fig 2 depicts a sample wireless sensor network, having a number of sensors and routers along with sink and external connection to internet. Consider Sa = {S0, S1, S2,……,Ska} is the set of active sensors and Sc={S0,S1,S2,……Skc} is the set of compromised sensor nodes. S=Sa U Sc. Also we are assuming that always Sa >Sc. It indicates that the numbers of active sensors are more in numbers than faulty sensors.

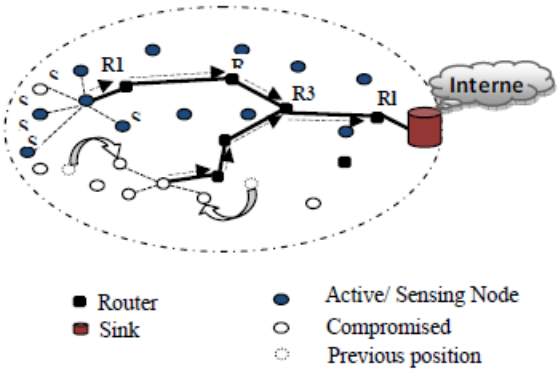

Fig 2 Sample Wireless Sensor Network with Gang Injection of False Data Attacks in Wireless Sensor Network

A. *Security Model*

zSecurity in Wireless Sensor Network is critical. In this security model, we propose a novel method for reducing the false data injection *[3]* from compromised nodes. Many methods are still in practices, but most of them cannot prevent gang injection of false data.

Fig 3 explains the architecture diagram, which shows how the proposed scheme filters false data and in effect how to identify compromised nodes.

B. *Resisting False Data Injection Attack*

In this attack model, the data shall be collided with the actual data which is sourced by the source node and the information shall be changed in to invalid or unknown information may not be understandable by the destination done by the adversary node. This would reduce the network performance in terms of throughput *[8]* and network availability. Security enhancing mechanisms are also provided with each sensed data. The general public key- private key cryptographic methods are adopted to maintain security for each data. Elliptical Curve Cryptography (ECC) is one of the oldest and strongest methods that still in practice. It is the main integral part of digital signatures also. TinyECC- A configurable Library for ECC is a version of ECC which is perfectly designed for wireless sensor network.

C. Algorithmic Approach

Each sensor node should be initiated well with the topology of the network or with the existing system. Each sensing result need to be filtered at each router to avoid false data report at the sink. The following are the algorithms used to evaluate each data report.

1) Sensor Node Initialization

The sink sets the public parameters as params= {E (IF_p),G,p,h()}. To initialize sensor nodes S={S0,S1,….Sk}, the sink invokes the Algorithm 1.

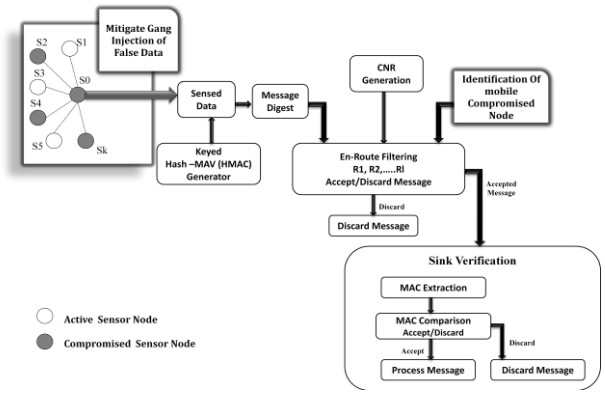

Fig 3 Architecture Diagram

Algorithm 1: Sensor Node Initialization

Procedure: SENSOR NODE INITIALIZATION
Input: params and sensor nodes (un initialized) S={S0, S1, S2….}
Output: Initialized sensor nodes S={S0, S1, S2…..}
For each sensor node Si

Preload Si with TinyECC, params and energy choose a random number xi in Zq*, the private key. And compute the public key Yi = xiG and install it in the Si

end for
return initialized S={S0, S1, S2…..}
end procedure

2) Reporting of Sensed Data

Sensed report will send to the sink via an established routing. The sensor node S0 has sensed some data m and is ready to report m to the sink via the routing path RS0 :[R1→R2→……..Rl→Sink.

Algorithm 2: CNR Based MAC Generation

ProcedureCNR BASED MACGENERATION

Input: params, Si€ (S_{S0}US0 m; T;R_{S0})
Output: Row_i
if Si believes the report m is true
then a neighboring node is assumed having the same ability to detect a true event as the source node and correctly judge the report m.
for j=1 to l do
mac_{ij}= MAC(m||T, k_{ij},1)
end for
mac_{is}= MAC(m||T, k_{is},∞)
else
for j=1 to l do
mac_{ij} is set as a random bit
end for
mac_{is} is set as a random string of length ∞
end if
Return Row_i = (mac_{i1}, mac_{i2},….., mac_{is})
end procedure

3) Intermediate Node Filtering

To filter the report at each node the following algorithm invoked. If the returned value is "*accept*", *Ri* will forward the message with timestamp and MAC to its downstream *[7]* node, Otherwise, (*m, T*, MAC) will be discarded.

Algorithm 3: MAC Verification

Procedure CNRBASEDMACVERIFICATION

Input: params, Rj ∈ {R1, R2,….,Rl}, m, T,S_{S0}
Output: *accept* or *reject*
Rj uses the non-interactive key pair establishment to compute shared keys with each node in {S0, S1, S2,….., Sk} as k_{0j}, k_{1j},…..,k_{kj}
set return value = "*accept*"
For *i* = 0 to *k* do

*macij = MAC(m||T, kij , 1)
if macij(XOR)*macij= 0
then
set returnvalue = "*reject*"
break
end if
end for
return returnvalue
end procedure

4) Sink Verification

If the *sink* receives the report (*m*, *T*, MAC), it checks the integrity of the message *m* and the timestamp *T*. If the returned value is "*accept*", the *sink* accepts the report *m*; otherwise, the *sink* rejects the report.

Algorithm 4: Sink Verification

Procedure SINKVERIFICATION
Input: params, k_{0s}, k_{1s},….., k_{ks}, m, T
Output: *accept* or *reject*
set returnvalue = "*accept*"
For *i* = 0 to *k* do
$*mac_{is} = MAC\ (m||T, k_{is}, \alpha)$
if $*mac_{is}(XOR)mac_{is}$= 0 then
Set returnvalue = "*reject*"
break
end if
end for
return returnvalue
End Procedure

5) Compromised Node Detection

False data sent by the compromised node can be detected and eliminated by the efficient detection mechanism *[5]*. The following is a part of the algorithm which detects efficiently the in active or malfunctioning node.
Input : *n*, number of iteration *Si*, seed *keyi*, encryption key for RC5
Output: *C* (*C*0, *C*1, . . . , *C*7), checksum value *C* is initialized to *keyi*; *tmp* is 1 byte initialized to 0; *j* is initialized to point to *C*0, the first byte of *C*;

for *i=1* to *n*/4 do
(*t*0, *t*1, *t*2, *t*3) = *RCkeyi* **5** (*Si*) ;
b = function of (*t*0, *t*1, *t*2, *t*3) mod *m*;
for *k=0* to *3* do
for *r=0* to *b-1* do
tmp = *tmp* $\oplus$ Mem[*tk* + r mod *m*];
Cj = *Cj* + *tmp*;
tmp = 0;
j = (*j* + 1) mod 8;

IV. IMPLEMENTATION

When come to the implementation, in simulation, the en-routing filtering probability can be tested as

$$\text{FPR} = \frac{\text{number of false data filtered by en-route nodes}}{\text{Total number of false data}}$$

In what follows, we provide the simulation results for FPR.

A. Simulation Settings

We study FPR using an NS2 simulator. In the simulations, 1000 sensor nodes with a transmission range R are randomly deployed in a certain interest region (CIR) of region 200 × 200 m2 interest region. Initially all experiments are done with 15 nodes and the average of en-routing filtering probabilities over all of these randomly sampled networks is reported.

Parameter	**Value**
Simulation area	200m × 200m
Number of sensor nodes	1000
Transmission range	*R* 15m, 20m
Compromised probability ρ	2%, 5%
# neighboring nodes *k*	4, 6
# routing nodes *l*	5,……,15

Table 1: Sample Simulation Setting

As the number of routing nodes increases, FPR increases. At the same time, by choosing more neighboring nodes involved in the protocol, i.e., the parameter *k* increases, FPR will further increase, even the compromised probability is 5%. Further observing the FPR with different transmission range *R*, we can see a relatively low FPR for *R* = 20m compared with that for *R* = 15m. The reason is that, under the same settings, when the transmission range increases, the number of compromised neighboring nodes will also increase, so the experienced and astute *A* has more chances to choose more compromised nodes participating in the attack to increase the success attack probability. Based

on these observations, we have the following theorem. Now let us see how we are going to prove this. We have the following relationship between FP and FPR, i.e.,

$FP = 1 - (1 - FPR)(1/2\infty)$

Where $1 - FPR$ is the success probability of injecting false attack escaping from the en-routing filtering, which consists of two parts:

i) FPA|*Sc*=*k*, the false positive probability when the number of participating neighboring compromised nodes $Sc = k$ in the attack

FPA|*Sc*<*k*, the false positive probability when $Sc < k$.

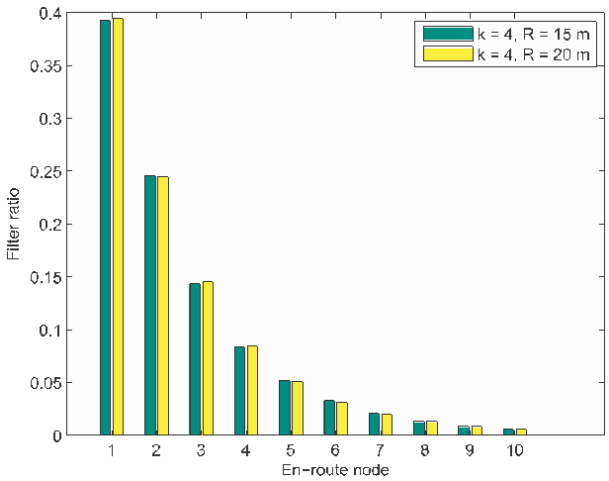

Fig. 4 Shows intermediate filtering Ratio at security parameter is 4

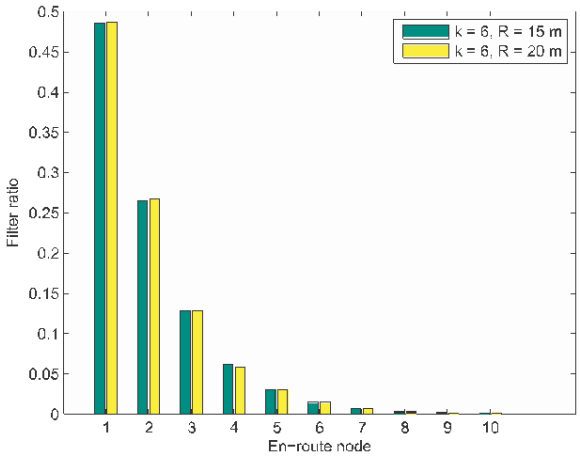

Fig. 5 shows intermediate filtering Ratio at security parameter is 6. As the parameter increases the filtering Ratio also increases.

V. PERFORMANCE EVALUATION

A Energy Consumption in Non-interactive Key pair Establishments

The *additional* computation costs of the scheme are mainly due to the expensive Elliptic Curve Diffie-Hellman (ECDH) operations during the non-interactive key pair establishments. The non-interactive key pair establishments are *averagely* distributed in each sensor node and only executed once during the routing establishment, the ECDH operation is not a heavy burden. When designing TinyECC based sensor node, we can choose a 160-bit elliptic curve for achieving the same security level as 1024-bit RSA. Assume that each sensor node is equipped with a low-power high performance sensor platform, i.e., MICAz. Then, according to this type of sensor platform only requires 50.82 mJ to establish a non interactive shared key.

VI. CONCLUSION

In this paper we are addressing a couple of security mechanisms that together filter the false data that are injected by compromised nodes. This scheme is an effective and efficient method to filter false data injected by compromised nodes and gang injection of false data. Rather than filtering the data entirely on the sink intermediate filtering strategy is added to avoid more traffic at the sink. It reduces unwanted energy conception at each nodes and wastage of time by forwarding each packet which contains false report.

VII. REFERECES

[1] Rongxing Lu, Xiaodong Lin, Chenxi Zhang, Haojin Zhu, Pin-Han Ho and Xuemin (Sherman) Shen, IEEE Communications- AICN: An Efficient Algorithm to Identify Compromised Nodes in Wireless Sensor Network 2008

[2] A. Liu and P. Ning, “TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks,” Proc. Seventh Int’l Conf. Information Processing in Sensor Networks (IPSN ’08), pp. 245-256, Apr. 2008

[3] Maurizio Adriano Strangio Department of Computer Science, Systems and Production University of Rome “Tor Vergata" Rome, ITALY 2005 ACM Symposium on Applied Computing Efficient DiffieHellmann TwoParty Key Agreement Protocols based on Elliptic Curves.

[4] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, “An Effective Key Management Scheme for Heterogeneous Sensor Networks,” Ad Hoc Networks, vol. 5, pp. 24-34, Jan. 2007.

[5] X. Lin, "CAT: Building Couples to Early Detect Node Compromise Attack in Wireless Sensor Networks," Proc. IEEE GLOBECOM '09, Nov.-Dec. 2009.

[6[Z. Benenson, C. Freiling, E. Hammerschmidt, S. Lucks, and L. Pimenidis, "Authenticated Query Flooding in Sensor Networks," Security and Privacy in Dynamic Environments, Springer, pp. 38-49, July 2006.

[7] X. Li, N. Santoro, and I. Stojmenovic, "Localized Distance- Sensitive Service Discovery in Wireless Sensor and Actor Networks," IEEE Trans. omputers, vol. 58, no. 9, pp. 1275- 1288, Sept. 2009.

[8] X. Li, A. Nayak, D. Simplot-Ryl, and I. Stojmenovic, "Sensor Placement in Sensor and Actuator Networks," Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication, Wiley, 2010.

◈◈◈