ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)

On 21st & 22nd March Organized by

K.L.N. College of Engineering , Madurai, Tamil Nadu, India

Hybrid Double Layered Security against Data Corruption and Node Compromise Attacks In WSN

K R Remesh Babu, Geethu K Mohan, Philip Samuel

Dept of Information Technology, Government Engineering College Painavu, Idukki, India.

Dept of Information Technology, Government Engineering College Painavu, Idukki, India.

Information Technology, Cochin University of Science And Technology, India

Abstract -- Wireless Sensor Network (WSN) is a collection of sensors that are of heterogeneous in nature. Data sensed from the environment are traversed through the network till it reaches the sink. The main focused problem of Wireless Sensor Network is the data integrity throughout the network. If the data is corrupted, considerable amount of energy is wasted at each time when the data is forwarded to the next node. The critical data corruption attack is done by compromised nodes. Various strategies have been introduced to identify the corrupted data and compromised node. This paper focuses a hybrid double layered security strategy for sensed data. The first step of security is applied by appending a Keyed Message Authentication Code (HMAC) to the sensed data by Secure Hash Algorithm (SHA-2/512) which is robust algorithm to ensure message security throughout the network. The second step of security is implemented by a modified form of ConstrAined Random Perturbation based pairwise keY (CARPY+) mechanism. In CARPY+ mechanism guaranteed key exchange between sender and receiver proves the sender nodes identity. Any fail while comparing the key which is extracted from the received message identifies the sender node as a malicious node. The proposed methodology improves the network performance by avoiding data corruption at the network layer and same time identifies the compromised nodes.

Key Words - Wireless Sensor Network, Compromised Node Attack, Data Corruption, Secure Hash Algorithm, ConstrAined Random Perturbation based pairwise keY(CARPY)

I. INTRODUCTION

A Wireless Sensor Network (WSN) consisting of heterogeneous sensors, that are capable of managing and privacy over the network. There are many techniques that are implemented for WSN to maintain the data integrity throughout the network. Most of the applications are implemented in static nodes. Since the advancement in the technology a static node application faces a difficult problem as its mobility. When we are introducing mobility to nodes, it is impossible to maintain a network topology. This paper introduces some technique to keep the data secrecy at the same time it identifies whether a node is compromised or not.

II. EXISTING SYSTEM

Wireless Sensor Networks are vulnerable to a variety of attacks like altered routing information, selective forwarding, sink hole, worm hole, attack on transit etc. By access, attacks are classified mainly into two categories. Routing attacks and attacks on transit. Data corruption is a type of attack on transit. Data corruption is either by compromised nodes or by attack on data in the network layer. Different techniques are adopted to reduce or to identify such attackers or to reduce such corrupted data. Detailed analyses on techniques are given in the literature survey below.

A. Literature Survey

Many approaches have been evolved for the packet filtration over the past few decades. One of an effective approach is Statistical En-Route Filtering of Injected False Data in Sensor Networks by Fan Ye, Haiyun Luo, Songwu Lu, Lixia Zhang (IEEE INFOCOM '04, Mar. 2004) [5] The main objective is to detect and drop false report. Same event is sensed by multiple sensors. Multiple MAC is associated with forwarding report. Each legitimate report carries multiple MACs generated by different nodes that detect the same stimulus.

Intermediate forwarding nodes detect incorrect MACs and filter out false reports. [1] At the sink level the report is tested once again so as to ensure the message integrity. Another important technique proposed was CAT: Building Couples to Early Detect Node Compromise Attack in Wireless Sensor Networks[4] by Xiaodong Lin (IEEE GLOBECOM '09, Nov.-Dec. 2009)[9]. This technique proposes a naval compromised node detection method. This is the first effort on addressing the node compromise problem in the first stage. This method presents a new couple-based scheme to detect the node compromise attack in early stage. Specifically, after sensor nodes are deployed, they first build couples in ad-hoc pattern. Then, the nodes within the same couple can monitor each other to detect any node compromise attempt. The major disadvantages of all the above technique are, the above, mentioned are applicable with static node and static sink. The proposed methodology focuses on mobile compromised nodes[6]. As an added advantage this method detects and drops false data as well as identifies compromised nodes also.

B. Problem Definition

Data corruption attacks [1] in Wireless Sensor Network are through compromised nodes and network layer data corruption. Any intruder can easily pass corrupted data through this nodes thereby they can drop the equilibrium of the sensor network [12]. Each node in a Wireless Sensor Network preloaded with identification key uniquely. When a node compromised, it is easier for the intruder to corrupt the messages, which in turn results wrong data interpretation and it produces false result. Sensed data is transmitted through the network, while forwarding each packet to the next node consumes considerable amount of energy as well as time.

Addressed Problem:

- Ensure Data integrity while sensing the data (Before sent to the network)
- Ensure entire message integrity throughout the network.

The main aim of this paper is to provide end to end security for data. Once data received in the sink it will process it directly. Wrong information will result in a faulty result. The available mechanism cannot be implemented easily because of high computation and storage issues of security keys. Also compromised node [4] identification needs much more complex algorithms. When come to the corrupted data identification many methods have been introduced. As an effective mechanism intermediate node filtering possesses high filtering capacity but it drains much energy [2].

III. PROPOSED SYSTEM

Each sensor node is deployed in the area of interest with specific keys to ensure the integrity of the nodes. The proposed methodology gives importance to both the data sensed from the sensor node and also the entire message that sending towards the sink. This is an effective mechanism which meets all the requirements of sensor nodes. This method successfully identifies large number of compromised nodes. This solution is explained in two sections.

- Giving security to the sensed data before sending.
- Giving security to the entire packet while sending to the network.

The implementation of security is done by two sections, which is explained below.

- Using Secure Hash Algorithm
- Modified ConstrAined Random Perturbation based pairwise keY (CARPY+) [2]

Advantages of the proposed methodology,

- Probability of presence of corrupted data in the network is less.
- Removal of malicious node increases the network performance.

A. Modified Random Perturbation based pairwise keY (CARPY+)

The ConstrAined Random Perturbation based pairwise keY (CARPY+) is derived from Blom's model. In modified CARPY+ scheme we are assumed [3] that of N nodes I = {S0, S1, S2…… S_N} In the CARPY+ method, defined the field Fq = {0, . . . , q−1},q > N, be a finite field. Every calculations are confined under a finite field called Fq, with security parameter 'λ'. It is possible to address λ compromised nodes. For a matrix G, we denote the element in the i-th row and j- th column of G by Gi,j , i-row of G by $Gi,\_$ and the j-th column of$G$ by $G\_,j$. Assume that a symmetric matrix $D \in$ F(λ+1)×(λ+1) q and a matrix $G \in$F(λ+1)×N q are randomly generated.

For each sensor node i, the row vector $Ai,\_$ and the column vector $G\_,i$ are stored in the node i. When two nodes i and j would like to have a common key, they exchange their columns of G in plaintext and then use their private rows of A to calculate Ki,j (= $Ai,\_ \cdot G\_,j$) and Kj,i (= $Aj,\_ \cdot G\_,i$), respectively. When D is totally known by the adversary [4], Blom's scheme becomes insecure. The communications become insecure after more than λ sensor nodes are compromised. The reason for this is that the row vector $Ai,\_$ in the sensor node i is directly related to the private matrix D.

To enhance security in Blom's key, we are adding a random noise to distort key. If the length of the key is 'l' then only the least r (r <l) bits of Blom's key are perturbed after the CRP is added. Thus, the first l-r bits of Blom's key are retained. If the desired key length is 'L', then CARPY+ have to execute l/(L-r) rounds to produce pair wise key. This is explained with an example given below.

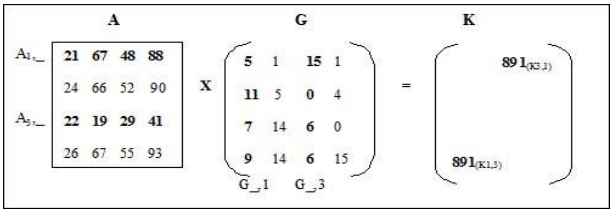

Fig. 1 Example of Blom's Scheme

To enhancing the security we are adding noise represented as ϕ, so that the addition of matrix with ϕ obtains the same matrix.

[20684888]=W1,_A1,+1=[2164888]+[_1100]

[20684888]=W3,_A3,+3=[22192941]+[_1_101]

ϕ_1 and ϕ_3 are the random noises for $W_{1,\_}$ and $W_{3,\_}$ respectively

$K'_{1,3} \neq K'_{3,1}$, But

X1,3=(11011)2=f10,5(876) and

X3,1=(11011)2=f10,5(884)

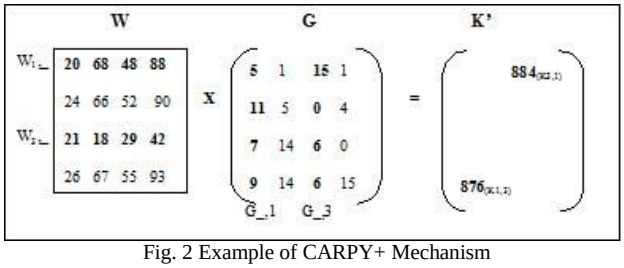

Fig. 2 Example of CARPY+ Mechanism

There are mainly two steps in CARPY+ scheme,

- *Off-line Step:* This step is used to find out the desired key length.
- *On-line Step:* This step is used to find out the pair wise key between two sensors after the node is deployed.

1) Off-line Step for CARPY+: Two main steps are involved in the off-line set up for CARPY+. They are mentioned below. [2]

Step 1: Select two matrixes randomly say 'D' and 'G' D(t)Fq(+1)(+1) and G(t)Fq(+1+(+1).

Step 2: Calculate $A^{(t)}$

$A^{(t)} = D^{(t)}. G^{(t)}$

Detailed algorithm for Off-line step for CARPY+

1. Calculate 'l'
2. T=1 to L/(l-r) // total number of rounds to be performed.
3. Generate $D^{(t)}$ and $G^{(t)}$. Calculate $A^{(t)}$.
4. For u=1 to N // for each sensor nodes.
5. Calculate $\phi^{(t)}_{Su}$ // Calculate CRP for each node.
6. Select one raw vector from obtained $\phi^{(t)}_{Su.}$
7. Calculate $W^{(t)}_{Su} = A^{(t)}_{Su} + \phi^{(t)}_{Su}$
8. Store $W^{(t)}_{Su,\_}$, $G^{(t)}_{Su,\_}$ into node 'Su'.

2) On-line Step for CARPY+: This is the second and most important step in CARPY+ mechanism. In this step the pair wise key between two node is extracted and compared where the key are identical or not [2]. The following is the steps to achieve the pair wise key.

Suppose Su and Sv wants to share the key. After executing 't' rounds of CARPY+,

Step 1: Calculation of Key

For sensor node Su and Sv calculates $K^{(t)}_{Su,Sv} = W^{(t)}_{Su,\_} * G^{(t)}_{\_,Sv}$ and $K^{(t)}_{Su,Sv} = W^{(t)}_{Sv,\_} * G^{(t)}_{\_,Su}$. After 't' rounds, the t^{th} part of the key is $X^{(t)}_{Su,Sv} = f\ l,r\ (K^{(t)}_{Su,Sv}) = f\ l,r(\ K^{(t)}_{Su,Sv})$. Where fl,r(x) is the most significant bit of l-bit binary representation of a number x.

Detailed Algorithm for on-line setup for CARPY+

1. From t=1 to £=L/(l-r)
2. Calculate $G^{(t)}_{\_,Sv}$
3. Calculate $K^{(t)}_{Su,Sv} = W^{(t)}_{Sv,\_} * G^{(t)}_{\_,Su}$
4. Calculate $X^{(t)}_{Su,Sv} = f\ l,r\ (K^{(t)}_{Su,Sv})$
5. Calculate $X_{Su,Sv} = X^{(1)}_{Su,Sv} \| X^{(2)}_{Su,Sv} \| \ldots\ldots \| X^{(£)}_{Su,Sv} \|$

The extracted key is compared with the stored key and if the two key are same then only S_u accept that message from S_v and hence it can be proved that the message is coming from an authorized sender. If the key comparison is not true this will indicate it as a malicious node and it will further prompted for successive testing for the confirmation of compromised node [7]. Let us explain the above steps via an example.

Example 1: Considering the above two diagrams, given that $q = 2^{10} - 3$, N = 4, λ = 3, r = 5, L = 5 and I = {1, 2, 3, 4}. The main idea of CARPY+ is shown in Fig. 2. In this example, l = 10 can be calculated. Since l−r = L, performing the CARPY+ scheme once is sufficient to generate a key with length L. Moreover, W1,_ comes from A1,_+ φ1,_, where A1,_ is shown in Fig. 1 and φ1,− is randomly chosen as a row vector [−1 1 0 0]. W3,_ can be obtained similarly by having [10] A3,_+ φ3,_, where φ3,_ = [−1 −1 0 1], as also shown in Fig. 2. Though $K'_{1,3} = W_{1,\_} * G_{\_,3}$ $W_{3,\_} * G_{\_,1} = K'_{3,1}$, their most significant l-r = 5 bits are the same, i.e., $X_{1,3} = f_{10,5}(228) = (000111)_2 = f_{10,5}(218) = X_{3,1}$. Hence, $X_{1,3}$ (= $X_{3,1}$) can be used as the pairwise key between sensor nodes with IDs 1 and 3 [2].

B. Secure Hash Algorithm (SHA) – 2/512

Secure Hash Algorithm (SHA) is an authentication technique. There are different situations where authentication is much important like masquerade, content modification, sequence modification and timing modification [8]. The main purpose for the use of SHA here is for masquerade [9]. In this kind of attack the corrupted data is inserted or data content is changed [1].

There are four different specifications of SHA. They are SHA 1, SHA 256, SHA 348 and SHA 512. One latest version of SHA 3 is released. The main advantages of using SHA 512 are,

- High probability of practical collision avoidance.
- Too difficult to broke the hash value.
- Message retrieval is almost impossible once it hashed.
- Even for the same message the hash value produced will be different [11].
- Highly robust in nature.

There are mainly two steps to generate a hash value. They are Preprocessing and Hash Computation. Detailed procedural explanation is given below [9].

1) Preprocessing: Step 1: Parsing the message:-

First convert messages in to 'm' bit of 'M' blocks. Secondly append '1' at the end of message bits followed by 'k' zero bits. Where 'k' is the smallest non-negative solution of the equation,

l+1+k=89mod1024---------- (1)

Total of 128 bit block will be achieved and total padded message will be multiples of 1028. That is N*1028

Parsed message =N*1028 ---------- (2)

Each message is represented as 64 bits and total of 16 blocks of 64 bits.

Step 2: Setting up of initializing values

Since it uses hexadecimal hash values 8 hash values have to be generated as $H^0_0, H^0_1, H^0_2, H^0_3, H^0_4, H^0_5, H^0_6$ and H^0_7. These values are the initial hash values which can be initialized with the hashing function.

2) *Hash Computation*

Step 1: Produces message Schedule:- The produced message schedule is of 80 constants of 64 bit length. Each hash values is of 64 bit length and finally 8*64 bit is generated [9].

Step 2: Iteratively generate values for hash function:- In the preprocessing step itself the first set of hash value is generated. By function the iteratively the next is generated.

IV. Implementation

The implementation in the real time is much costlier, hence this is done ad a simulation to identify various scenarios in the identification of compromised nodes as well as the message security. This can be much explained in another way. That is the number of messages that have been rejected or dropped without forwarding. The corrupted message is filtering is explained in the section III A.

The filtering probability ratio FPR can be calculated by,

FPR = number of false data filtered by en-route nodes

Total number of false data

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

A. Simulation Settings

We are simulating the experiment in NS2 environment. 1000 sample nodes are taken for experiment. We fix the transmission range as R in a certain interest region (CIR) of region 300 × 300 m2. Initially we start with 15 nodes [4].

V. Performance Evaluation

A Energy Consumption in Non-interactive Key pair Establishments

To analyses the total energy consumption we have to consider both the energy consumption for CARPY+ and for SHA. Both analyses are explained separately. Firstly let us find out the energy consumption CARPY+ mechanism only.

CARPY+ [2] scheme provide high level of security to the data at the same tome identified the compromised node. While calculating the energy consumption, consider the energy spend for communication and also for computation[6]. That includes energy for encryption (e_e) and decryption (e_d) in computation and energy for receiving (e_r) and transmitting (e_t) in communication.

B. Energy consumption for Key Establishment (For Both off-line and on-line)

Random perturbation is the only method which reduces the energy consumption at the same time increasing the security and reduces the communication overhead. The energy for Random Perturbation (RPB) is given below,

$$E_{RPB} = E^{Comm}_{RPB} + E^{Comp}_{RPB} \text{ ---------- (3)}$$

To deploy the keys in the sensor node it takes a onetime energy loss.

$$E^{Comm}_{RPB} = h*((e_t/1-P_{lossr}) + e_r) *(L_m/L_{packet}) \text{ ---------- (4)}$$

Where L_m is the length of hash (in bytes)

$$E^{Comp}_{RPB} = £ * \lambda (e_a + e_m)+ (£* e_e) + (£-1) \text{ ---------- (5)}$$

C. Analysis of Various Matrices

An analysis of existing system versus proposed CARPY+ scheme is illustrated in the following graph. Explanation for each graph is mentioned below.

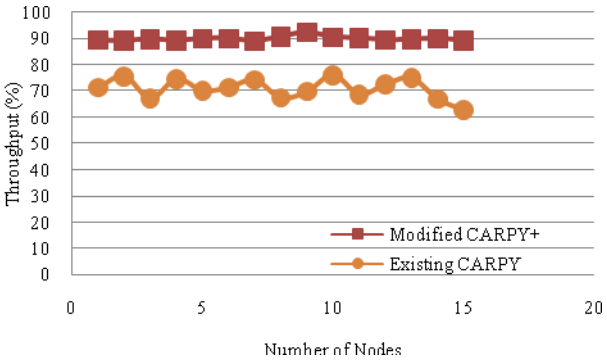

Fig. 3 Number of node versus average throughput

Figure 3 depicts average throghput against total number of nodes. A relative analysis is done here with the proposed CARPY+ scheme. The graph plots a noticable change in throghput.

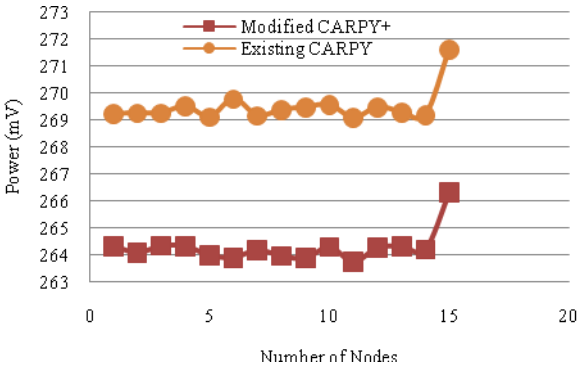

Fig. 4 Consumption

An analysis of nodes against power consumption for individual nodes ploated in Figure 4. After applying CARPY+ scheme, power consumption in individual nodes are reduced .

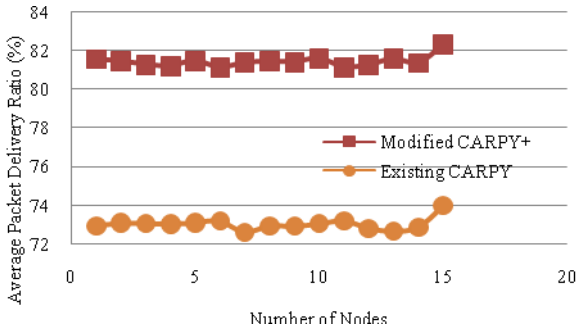

Fig. 5 Average packet delivery ratio.

Packet delivery ratio, that means the total packet received out of send also increase due to low corruption and node compermises. A graphical representation for the same is picturised here in Figure 5.

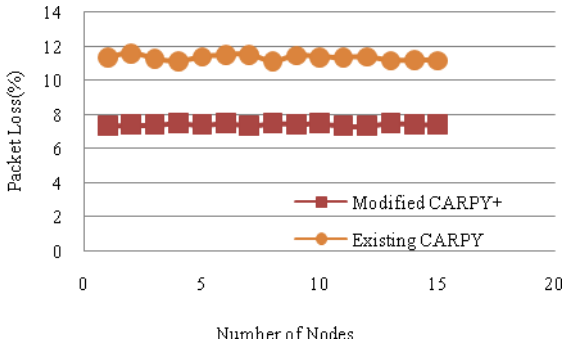

Fig. 6 Packet loss ratio

Gradual degradation in packet loss in each node is visible in Figure 6. Since CARPY+ scheme can provide security to both data and node. Hence the probability of corrupted data existance is merely negligible.

VI. Conclusion

This paper addresses a couple of security mechanisms that together detect corrupted data and there by compromised node also. 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, hence it reduces energy wastage in each node. To confirm the node compromising feature a software attested code base testing can add as a future work.

References

[1] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Xiaohui Liang, Xuemin (Sherman) Shen, "*BECAN: A Bandwidth-Efficient Cooperative Authentication Scheme for Filtering Injected False Data in Wireless Sensor Networks*" IEEE Transactions On Parallel And Distributed Systems, Vol. 23, No. 1, January 2012

[2] Chia-Mu Yu, Chun-Shien Lu, Sy-Yen Kuo, "*Non interactive Pairwise Key Establishment for Sensor Network*" IEEE Transactions on Information Forencis and Security, vol. 5, no. 3, Sept. 2010.

[3] 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.

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

[5] 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.

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

[7] 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

[8] 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.

[9] "Federal Information Processing Standards Publication" 180-2, 2002 August 1 Specifications for the SECURE HASH STANDARD,http://csrc.nist.gov/publications/PubsFIPS.html #fips180-4.

[10] D. Djenouri, L. Khelladi, A. N. Badache, "A survey o f security issues in mobile ad hoc and sensor networks", IEEE Communications Surveys & Tutorials 7 (4) (2005) 2–28.

[11] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constraints and Approaches for Distributed Sensor Network Security," NAI Labs, Tech. Rep. 00-010, September 2000.

[12] Mo, Yilin Garone, Emanuele; Casavola, Alessandro; Sinopoli, Bruno, "False data injection attacks against state estimation in wireless sensor networks",Decision and Control (CDC), 2010 49th IEEE Conference on Date of Conference: 15-17 Dec. 2010.