Secure System Practices and Data Access Management in Wireless Sensor Network

A. R. Uttarkar
Student, ME Computer Engineering
JSPM's JSCOE, Pune
Maharashtra, India.

H. A. Hingoliwala
Associate Professor
Department of Computer Engineering
JSPM's JSCOE, Pune
Maharashtra, India

ABSTRACT
Wireless Sensor Networks (WSN) are a heterogeneous system with a collection of sensors distributed unbalanced patterns in remote areas, and often in unfriendly environments, without any pre-deployed architecture, and with limited hardware inside them. As the use of wireless sensor networks continuously growing, it should require efficient security mechanisms. Therefore to ensure the security of communication and data access control in WSN plays a vital role and has top significance. Because sensor networks may interact with sensitive data and operate in hostile unattended environments, it is necessary that these security issues should be addressed from the beginning of design of the system itself. In this paper we are presenting secure network protocol and security mechanism for Data Access Control which is built upon network layer of WSNs and our focus is on data access control and secure network protocol. Here Virtual Counter Manager (VCM) along with the synchronized incremental counter is presented for detection of replay and jamming attack using basis of symmetric key cryptosystem. For access control & prevention from unauthorized access we are presenting Key-Lock Matching (KLM) method.

General Terms
Wireless Sensors, Network Topology, hashing, synchronization

Keywords
Attacks, KLM, Sensor Networks, VCM

1. INTRODUCTION
Wireless Sensor Networks (WSNs) are widely used in many commercial applications , military services, industrial research and various medical science applications. Since there are the limited resources in sensor nodes in the wireless environment, these types of networks require special security requirements besides to the security needs in traditional networks. Wireless Sensor Networks are basically heterogeneous systems which are containing many no of small devices called sensor nodes and actuators having ability of general-purpose computing. Range of Sensors in WSN may vary from few hundred to thousands. These sensor nodes will have limited resources of power, storage, communication and processing capabilities [2] [3] [4]. Since sensor nodes may collect sensitive information security and privacy of nodes become an important issue in large number of nodes WSNs [1]. Because sensor nodes are resource limited nodes our conventional security mechanisms are not suitable for WSNs, therefore special security methods are need to be implemented.

2. LITEATURE SURVEY
It's not a case that security mechanism suitable for WSNs had yet not constructed, there are various security methods like SPIN [5], TinySec[6], ZigBee[7] and MiniSec[8]. TinySec mechanism achieves low energy consumption by reducing some level of security provided in system whereas ZigBee suffers from high energy consumption. SPIN technique uses synchronization between sender and receiver in system. MiniSec achieves low energy consumption by appending a few bits of the IV to each packet [1]. Another works focuses on secure network protocol and there is no consideration for security of data stored inside the node. Methods used by Kun et al's [9] guarantee that all traffic in system is authenticated but it is unable to look after or detect replay or jamming attack. Recent technologies are developed for secure data storage for social or cloud networks as well as for sensor networks for maintain or preserving privacy. Instead of privacy preservation in network our focus is on authorization of data access stored in sensor node. Here we are proposing a secure network protocol for wireless sensor networks which works with low energy consumptions as well as establishes high security mechanism on sensor nodes. It provides a secure network protocol to permit data transmitted in an encrypted format in air and a filtering capability to permit or deny data access based on some set of rules used for protecting data from illegal access. The design is based on existing Authenticated Encryption Standard (AES) which is most suitable block cipher for WSNs [10] [11]. Virtual Counter Manager (VCM) with synchronized incremental counter is used for resisting replay/jamming attack. Memory Data Access Control Policy (MDACP) is presented along with Key Lock Matching (KLM) method to achieve memory data access control. In KLM, each user is coupled with a key like a prime number and each file is associated with a lock value. For each file, there are some corresponding locks, which can be extracted from prime factorization. Through simple computations on the basis of keys and locks, protected memory data can be accessed. Here, data access control is designed exclusively for function nodes [1].

The rest of the paper is organized as follows. Section III describes the system topology model and attack model. Section IV describes the details for proposed method. Section V describes system model with corresponding algorithms. Section VI includes experimental results and future scope is discussed in VII. Finally in section VIII we are concluding remarks.

3. SYSTEM TOPOLOGY AND ATTACK MODEL

3.1 System Topology

Whatever is our application, the network topology plays a key role in determining the quality of WSNs since it affects both the sensing capability and the wireless connectivity. Here relationship between the nodes is illustrated as in Fig. 1. There are three types of nodes, includes Leader Node (LN), Function Node (FN), and Sensor Node (SN), in our sensor network topology. They are classified according to their hardware resources with conditions like remaining energy, memory size etc. The network region is partitioned into various physical clusters, each of which contains a FN having charge of SNs in that cluster.

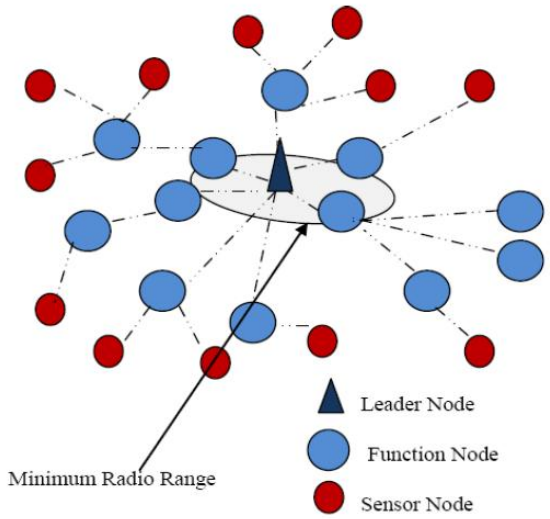

Figure 1: Network Topology Used

Depending on existing applications, clusters may overlaps with each other and hence Sensor Nodes in overlapping may be associated with multiple Function Nodes. In each individual cluster, Sensor Nodes are responsible for collecting sensed data, while Function Nodes will aggregate the data from their associated Sensor Nodes. Function Nodes can also send commands to Sensor Nodes to keep utility data, appliances, etc. in inside memory. They forward the received data to their upper level nodes that is to Leader Node. The Leader Node is a network owner with ample resources that can query data by an on-demand wireless link connected to all Function Nodes. To prevent storage overflow of Function Nodes, the Leader Nodes can also be periodically dispatched to collect data and empty the storage of Function Nodes.

3.2 Attack Model

The opponent can launch both external and internal attacks. In external attacks, the opponent does not control any valid nodes in the network. Instead, he may attempt to tap sensitive information, inject fake messages, replay previously intercepted messages, and pretend to be valid sensor nodes. Moreover, we assume that the opponent can jam the communication between two nodes by transmitting signals that disturb packet reception at the receiver. The opponent may also launch DoS attacks by, for example, false data injection or path-based DoS (PDoS) to reduce the energy of Function Nodes. As for internal attacks, we do not consider that the Function Node will be captured. Instead, we consider that the opponent may attempt to read the data stored in Function Nodes memories, utilizing an unauthorized node to read important data from FNs randomly

3.2 Key Distribution

To keep the confidentiality of messages transmitted over the network, there are two types of keys used in our system:

Session keys: Used for Leader/Function Nodes to broadcast packet to Function/Sensor Node

Pair wise keys: Used for each pair of nodes.

Session key is distributed in advance before deployment of sensors in network. After sensor deployment, pair wise keys are constructed for pairs of sensor nodes by applying our CARPY+ scheme [12]. The main advantage of CARPY+ is that it can establish a pair wise key between each pair of sensor nodes without needing any communication. This property is essential in constructing the Constrained Function based Authentication (CFA) scheme, because establishing a key via communication incurs an authentication problem, leading to circular dependency. CARPY+ is also flexible to a large number of node compromises so that the complexity for breaking the CARPY+ scheme is Ω (2l+1), where l is a security parameter independent of the number of sensor nodes. When updating the session keys, we customize stateless session keys update schemes, which organize one-way key chain to facilitate the authentication of future keys based on previous ones. In stateless session keys update scheme, network owner α uses the pair wise key K α, β shared with each non-revoked node β to encrypt the new session key.

4. PROPOSED METHOD

The structure of proposed protocol stack will be as like shown in Fig 2. There are two lowest layers hardware and hardware abstraction Layer like device drivers. These two layers are responsible for providing all basic services and components. TinyOS resides on the top of these two lower layers. Protocol is constructed within TinyOS layer. It includes different things like Memory Data Access Control Policy, Event Handler, VCM, Query Logic and key pool. We use a symmetric key cryptosystem i.e. AES-Authentication Encryption Standard with a communication key session key or pair wise key to encrypt the data for the purpose of data confidentiality. The False data injection attack and path based denial of service (PDoS) attack can wear out limited energies of FNs and possibly black out a section of the monitored area. To handle this it is necessary that some authentication mechanism should be there for preventing the communications in the network from DoS attacks. There have been many authentication schemes proposed for wireless sensor networks [1]. But Constrained Function based Authentication (CFA)[13] is only scheme of authentication which is supporting en-route filtering with only single packet overhead.

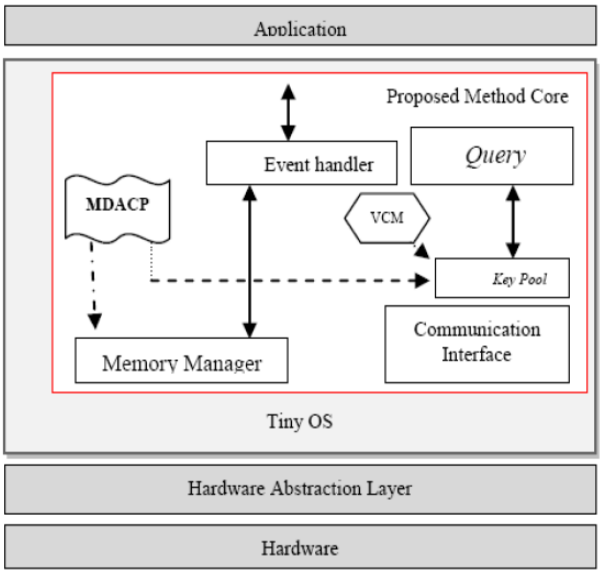

Figure 2: Proposed Method Protocol Stack

In the CFA scheme, the network planner, before sensor deployment, selects a secret polynomial f(x, y, z, w) from the constrained function whose coefficients should be kept secret. For simplicity, we assume that the degree of each variable in f(x, y, z, w) is the same, although they can be distinct. For each node u, the network planner constructs two polynomials, fu,1(y, z, w) = f(u, y, z, w) and fu,2(x, z, w) = f(x, u, z, w) Since directly storing these two polynomials enables the opponent to obtain the coefficients of f(x, y, z, w) by capturing a few nodes, the authentication polynomial authu(y, z,w) and verification polynomial verfu(x, z, w) should be constructed from the polynomials fu,1(y, z,w) and fu,2(x, z ,w), respectively, by adding independent perturbation polynomials. Afterwards, the authentication and verification polynomials are stored in node u. For source node u, the MAC attached to the message msg is calculated according to its own authentication polynomial. Let the verification number be the result calculated from the verification polynomial verfu(x, z,w) by applying the claimed source node ID, the shared pair wise key, and the hashed message into x, z, and w, respectively. The receiver considers the received message authentic and unbroken if and only if the verification difference, which is the difference between the MAC and its calculated verification number, is within certain predetermined range. CFA is slightly modified and incorporated with AES in Offset Code Book (OCB) mode within our proposed method to provide DoS resilience In order to incorporate CFA in our system, AES in OCB mode takes msg, Kv,u, and an IV as inputs, and generates the cipher text E Ku,v,IV (msg) and hash value h(msg). It should be noted that the pair wise key Ku,v = Kv,u is constructed by employing our CARPY+ scheme[12] on nodes u and v, respectively.

Our proposed method uses a synchronized incremental counter as an Initialization Vector (IV) for achieving semantic security. Specifically, the IV associated with a buffer filter is used to detect replay and jamming attacks instead of appending IVs into packets transmitted in the air. With the synchronized incremental counter, we construct a VCM within each node for initializing the counter and maintaining counter synchronization between the sender and receiver. The synchronized incremental counter in each node increases one count per average delay automatically. Also, we define the maximum counter synchronization error (MCSE) to be an experiment based delay counter, δ, between any pair of nodes. In other words, when the packet transmission time is much longer than δ, the jamming attack can be detected at receiver. If a packet does not suffer the jamming attack, the receiver applies a buffer filter to detect whether the packet suffers the replay attack.

5. SYSTEM MODEL

Message Transmission: When u wants to send message (msg) to destination node v, it calculates Message Authentication Code

MACu(v,msg)=authu(v,K u,v, h(msg))+Nu,s;

Where Nu,s is used for perturbation and h (msg) is value that is generated based on AES in OCB mode. The packet M with header is send to v possibly through multipath.

Message Verification: At receiving side of packet M destination node calculates verification number according to his own verification polynomial VDv,u

VDv,u= | verfu(u,Kvu,h(msg))- MACu(v,msg)|

If VDv,u€ [0, 2r-1 -1] then authenticity and integrity of packet M is successfully verified , otherwise packet M will be discarded .

Verification process for intermediate node is same as the destination node.

The synchronized incremental counter approach at sender side is as per given in algorithmic steps. It is assuming sender has started to send packet to receiver. Sender gets counter value used as an IV from VCM.

Algorithm: 1

Sender side: Synchronized Incremental Counter Approach

- Scenario: Node u sends message msg to node v.
- Input: IV from sender VCM and Ku,v
- Output: Packet processed via AES-OCFA

1. If radio channel=success then
2. Send out packet
3. Else
4. Back off for random period of time and

   then go to the step 1
5. End.

After some propagation delay in air receiver node will receive an incoming packet. It will perform two activities:

1. Determination of whether packet is legitimate one.
2. Determination of whether packet has suffered attacks.

Algorithm 2 first checks whether the packet had suffered from Dos attack. If yes, then it means packet does not suffered from DoS attack, after words packet is checked whether replayed or jammed. In next step receiver gets a current counter value from virtual counter manager and calculates range counter value. Range counter interval is a set of IVs to verify received packet. If all decryption fail within the interval defined in range counter interval then packet may be jammed or invalid. Hence packets are dropped. In order to detect replay attack,

simply buffer is used to filter out duplicate packets. For this purpose receiver queries the corresponding buffer filter for tuple of packet. If it is returning successful then it means to no duplicated tuples, and therefore packet is considered for adding into buffer filter.

Algorithm: 2

Receiver side: Synchronized Incremental Counter Approach

- Scenario: Node v receives Packet M from node u.
- Input: Range Initialization Counter
- Output: verification result.

1. If packet not suffered from DoS attack==TRUE
2. Compute Range Counter Interval(RCI)
3. If decrypt all ≠success

   then Drop packet
4. Otherwise check Buffer filter
5. If Source Address and RCI matches

   then store in Buffer Filter

   otherwise discard packet
6. End

5.1 Counter Synchronization

At the start, all nodes boot up with the same counter value. When the network runs for a period of time, the counters of nodes may lose synchronization. Recent advances in secure sensor network time synchronization [14] enable pair wise time synchronization with error of mere μs. Transmission delay between neighboring nodes are on the order of ms. Thus, we launch VCM to synchronize counter value based on Secure Pair wise Synchronization (SPS) protocol [14]. Here node A sends synchronization packet to B at clock C1 and node B receives the corresponding packet at C2. At clock C3 B sends acknowledgement packet which contains values C2 and C3. When node A receives the packet at C4 it calculates end to end counter delay. Thus jamming attack is detected through comparison of Cd with δ. In the proposed PCS algorithm integrity and authenticity are ensured through the use of MAC. This prevents external attackers from successfully modifying any values in the synchronization process. Furthermore, the opponent cannot pretend to be node B as it does not know the secret key KA,B. Replay attacks are avoided by using an IV during the handshake.

5.3 Memory Data Access Control Policy

To defend against unauthorized users access of data we are apply MDACP. The personal information, key material and other important information will be encrypted using AES-OCFA and stored in inside memory. In MDACP, each user is associated with a key like a prime number and each file is associated with a lock value. For each file, there are some corresponding locks, which can be extracted from prime factorization. Through simple computations on the basis of keys and locks, protected memory data can be accessed. An MDACP stores encrypted file in nodes as well as it binds user keys and specific encrypted files together. Due to this it reduces the risk factor of comprising keys by various attacks. Additionally, by employing the KLM method, whenever a new user or file is joined, the corresponding key values and lock values will be determined immediately without changing any previously defined keys and locks. This scalability characteristic motivates us to employ KLM for the design of MDACP.

5.2 Software Requirement Specification

The performance analysis of system is made on Windows based platform under Java Universal Grange Framework. Java software development kit with minimum 1.5 versions or higher and eclipse/net beans IDE is used for simulating the system. Nodes are simulated using the Graphical representation in Java through AWT and swing based classes and using event handling. Our analysis focus on replaying and jamming attack detection, resilience against node capture attack, evolution of minimum time required for pair wise counter synchronization, semantic security and data access control and energy consumption.

5.4vComparison of Our Method with Some state of the art methods

System proposed here is a fully – implemented general purpose security mechanism for a WSN. The comparison among some of well known methods and method proposed here is shown in table 1 and table 2.

Table I and Table II

Comparisons of Our Method with Some State-of-The-Art Methods (N: Number of Nodes; Φ: Packet Loss Rate; I: Bytes of The IV)

Method Used	Replay Detection	Jamming Detection	DoS Resilience
SPIN[5]	YES	YES	NO
TinySec[6]	NO	NO	NO
Zigbee[7]	YES	NO	NO
MiniSec[8]	YES	YES	NO
Proposed Method	YES	YES	YES

Method Used	Memory Access Control	Packet Security Overhead	Communication Cost
SPIN[5]	NO	Counter resynchronization	$O(N\Phi^c)$
TinySec[6]	NO	With 8 –bit IV	O(N+NI)
Zigbee[7]	NO	With 8 –bit IV	O(N+NI)
MiniSec[8]	NO	Few bit of the IV	O(N+NI)
Proposed Method	YES	NO	O(N)

The communication overhead is analyzed in terms of single packet transmitted between the sender and receiver and in terms of traffic overhead for packets transmitted in network. The results for analysis is shown in table III and table IV

Table III and Table IV

Comparisons of Communication Overhead

Method	Total Size(Bytes)	Energy (mAs)	Increase over TinyOS(%)
TinyOS	40	0.03776	0
SPINS	45	0.04246	12.5
TinySec	48	0.04528	20
MiniSec	43	0.04058	7.5
Proposed Method	40	0.03776	0

Method	Payload (bytes)	Packet Overhead (Bytes)	Security Overhead (Bytes)
TinyOS	28	12	0
SPINS	28	17	5
TinySec	28	20	8
MiniSec	28	15	3
Proposed Method	28	12	0

6. FUTURE SCOPES

For a large-scale network, strategies such as Bloom Filter [15] may be useful in reducing the storage overhead. This issue will be further studied in the future. The proposed algorithm will contain 4 steps and make use of bloom filter technique in its implementation.

Step I] Initialization: Activate sensor nodes on events (e1, e2, e3 ….en).Sink node broadcast to all authenticated task message and Nonce.

Step II] Event chaining and Aggregation : When node u detects event Nei, it generates bloom filter

BFu,ei=HMAC(Ku, idu||Nei)

which include raw message by node u to claim event Nei

For Aggregation if aggregator node v receives messages of Bfiu where i = 1,2,3,4,…….j

Aggregated Message= BFi =BF1 U BF2 U BF3 …….

Step III] Processing Reports: Base station l reports BFj using distinct keys shared with nodes.

Step IV] The sink broadcast l reports to whole network. The sink will notify all nodes to update the event identifier with Nonce to protect replay attack.

7. CONCLUSION

The method proposed here, is implemented on Windows Java Universal Grange Framework. It is an efficient network layer security system and is the fully implemented security mechanism that provides protection for both inside memory data and outside network message. It is achieving the goals of much less energy consumption and higher security than previous works in this area. Along with this it provides flexibility of deploying system with lower cost and higher security platforms

8. ACKNOWLEDGMENT

I express true sense of gratitude towards my project guide Prof. H.A. Hingoliwala, Associate Professor Computer Department for his invaluable co-operation and guidance that he gave me throughout my project. I specially thank our P.G coordinator Prof. M. D. Ingle for inspiring me and providing me all the lab facilities, I would also like to express my appreciation and thanks to HOD Prof. S.M. Shinde & JSCOE Principal Dr. M.G. Jadhav and all my friends who knowingly or unknowingly have assisted me throughout my hard work.

9. REFERENCES

[1] Yao-Tung Tsou, Chun-Shien Lu, " MoteSec-Aware: A Practical Secure Mechanism for Wireless Sensor Networks" IEEE transaction on Wireless Communication Vol 12, No.6 June 2013.

[2] David E. Culler, Wei Hong, "Wireless Sensor Networks: Introduction" Communications Of The ACM, June 2004/Vol. 47, No. 6, pp. 30-33.

[3] Adrian Perrig, Robert Szewczyk, J.D. Tygar, Victorwen and David E Culler, "SPINS: Security Protocols for Sensor Networks" Wireless Networks 8, 521–534, 200 ©□ 2002 Kluwer Academic Publishers. Manufactured in the Netherlands.

[4] "Key Distribution Mechanisms for Wireless Sensor Networks: A survey", TR-05-07, Department of Computer Science, Rensselaer Polytechnic Institute.

[5] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: security protocols for sensor networks," in Proc. 2001 International Conference on Mobile Computing and Networking, pp. 189–199.

[6] C. Karlof, N. Sastry, and D. Wagner, "TinySec: link layer Security architecture for wireless sensor networks," in Proc. 2004 International Conference on Embedded Networked Sensor Systems, pp. 162–175.

[7] ZigBee Alliance, Zigbee specifications, Technical Report Document 053474r06, 2005.

[8] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: a secure sensor network communication architecture," in roc. 2007 International Conference on Information Processing in Sensor Networks, pp. 479 - 488.

[9] S. Kun, L. An, N. Peng, and M. Douglas, "Securing network access in wireless sensor networks," in Proc.2009 International Conference on Wireless Network Security, pp. 261–268.

[10] L. Casado and P. Tsigas, "Contikisec: a secure network layer for wireless sensor networks under the Contiki operating system," in Proc. 2009 Nordic Conference on Secure IT Systems, pp. 133–147.

[11] NIST, National Institute of Standards and Technology, Computer Security Division, AES standard. Available: http://csrc.nist.gov/archive/aes/index.html, 2001.

[12] C.M. Yu, C. S. Lu, and S. Y. Kuo, "Non-interactive Pair wise key establishment for sensor networks," IEEE Trans. Inf. Forensic and Security, vol. 5, no. 3.2010

[13] C. M. Yu, Y. T. Tsou, C. S. Lu, and S. Y. Kuo,"Constrained function based message authentication fornsensor networks," IEEE Trans. Inf. Zorensic and Security, vol. 6, no. 2, pp. 407–425, 2011.

[14] S. Ganeriwal, S. Capkun, and M. B. Srivastava, "Secure time synchronization in sensor networks," ACM Trans Inf. and Systems Security, vol. 11, no. 4, pp. 1–35, 2006.

[15] H. Burton, "Bloom: space/time trade-offs in hash coding with allowable errors," Commun. of the ACM, vol. 13, no. 7, pp. 422–426, 1970.