The Cryptographic Security Solution and Distributed Reprogramming Approach for Wireless Sensor Networks: A Brief Survey

Chandramouli H Lavanya Chintapatla Dr. Somashekhar C Desai Dr. Prakash S
Associate Professor, Dept. Of CS & E, East Point College of Engg. & Tech., Bangalore, India
M. Tech Scholar (SE), East Point College of Engg. & Tech., Bangalore, India
Professor, V.P Dr P.G Halakatti College of Engg. & Tech., Bijapur,India
Pofessor & Head, Dept. Of CS & E, East Point College of Engg. & Tech., Bangalore, India

Abstract

The Security is the predominant focus of a Wireless Sensor Network. Having a secure and reliable WSN ensures the success of the product as a whole. This paper introduce two solutions, a centralized approach and A'SDRP. Each of these solutions has been developed to satisfy the reprogramming needs of the WSN along with a medium for secure data transfer. This paper talks about these two protocols in detail. A'SDRP is more scalable and is considered to be better than the centralized approach, as in this case multiple users can reprogram the sensor nodes without having to involve the base station and which is sometimes desirable and reduces the dependency on the base station. A'SDRP introduces a new set of capabilities such as assigning different reprogramming privileges per user compared to the centralized approach.

1. Introduction

A wireless sensor network consists of very small nodes that are deployed in some geographic areas. Sensor networks are used for object tracking, surveillance, seismic data collection and reporting, and many other applications. Since this data from the sensor networks are communicated wirelessly, the data should be less prone to attacks and authenticated before viewing. This makes security of these WSN a necessity for the success of these solutions. In section 2 of this paper security requirements, threats, and few solutions are discussed.

To secure the WSNs, there are many approaches; one of them introduced is a Centralized security approach which uses Secret Key (SK) cryptography and key management along with re-keying support for secure reprogramming sensor nodes and to reduce the communication and storage requirements of each node. The other approach is a Advanced (or Modified) Secure and Distributed Reprogramming Protocol (A'SDRP) approach where every user has his own privileges and therefore extending the applications and scalability of the network.

2. Security Concerns with WSN

In order to have a reliable and secure wireless sensor networks we need the following security components:

2.1 Data confidentiality or authentication

A WSN should be able to keep information hidden from unauthorized parties, and the nodes within the network should be able to identify genuine data and messages from unauthorized parties should be ignored. Messages to configure and control some of the nodes in WSN should be accepted from authentic sources only and mischievous sources should not be able to affect the working of the WSN. In case of using WSN for surveillance for military, this requirement becomes a necessary condition [3].

2.2 Integrity and availability

Data once generated by a sensor node should be received by its destination node or receiver, or in another work should be available and this data should not be corruptions by any adversaries [7].

2.3 Freshness

Freshness means that the data is recent, and it ensures that the adversary has not replayed old data [3].

In today's WSN security threats can be classified into following ways, brifly mentioned below:

2.4 Denial of service

Denial of Service (DoS) is formed by the accidental crash of nodes or due to malicious action. In WSNs, a large number of DoS attacks are possible across different layers. In physical layer the DoS attacks could be jamming and tampering, at link layer collision, at network layer, neglect and greed and at transport layer this attack could be performed by malicious flooding and de-synchronization [8].

2.5 Sybil Attack

A Sybil attack [3] is an attack in which an attacker destabilizes the reputation scheme of a peer-to-peer network by creating a huge number of pseudonymous entities, using them to gain a disproportionately big influence.

2.6 Black hole / Sink hole Attack

Sinkhole attack means, a malicious node acts as a sinkhole [4] to magnetize all the traffic in the sensor network.

2.7 Wormhole attack

In Wormhole attack, the attacker records the bits (or packets) at individual location in the network and subway those to another location [4][8].

3. Security Solutions and Approaches for Wireless Sensor Ntework

In the following sections security solutions and approach like secret key cryptography based security are discussed.

3.1 Security Solutions for WSN

In general there are several concepts that are available for WSN security today. They are broadly described as:

3.1.1 Encryption

This is a very basic solution. Techniques such as message authentication codes, symmetric key encryption schemes and public key cryptography are used to avoid unauthorized devices to eavesdrop or even add messages into the network [7].

3.1.2 Shared Keys

Key establishment (re-keying) is completed using one of many public-keys.

3.1.3 Protected Grouping

Sensor nodes are required to bind themselves in order to complete a particular task it is important that the group members communicate securing each other [10].

3.1.4 Secure Data aggregation

To decrease overhead cost and network traffic, sensor node aggregates measurements before sending them to the base station [3][7].

3.1.5 SPINS

Security Protocols for Sensor Networks: SPINS optimized for resource constrained environments and Wireless communication [11].

3.1.6 TinySec

Link layer security Architecture: Sensor network applications consists TinySec, which is lightweight, general security package [11].

3.1.7 Secret key Cryptography

The key is embedded in the source code of every node to protect the other keys in its non-volatile memory [3].

3.1.8 SDRP

The aim of this protocol is to secure the reprogramming and to reduce the communication overhead and storage requirements of each node [6].

3.2 Secret Key Cryptography Based Security Approach for WSN

This is the centralized approach based security on WSN. This approach ensures that we achieve the following goals:

3.2.1 Backward Secrecy

Even if an adversary recovered an adjacent subset of keys, it is impossible to recover the previous keys.

3.2.2 Privacy

Even the node is physically captured by an adversary; the secret information in the node's memory cannot be retrieved.

3.2.3 Data Integrity

Data Integrity ensures that the data during transmission over the network is not modified by an adversary.

3.2.4 Secure Management

Our mechanism provide secure method for key generation as well as for re-keying which is very much necessary in defending against cryptography attacks [13].

To achieve the goals mentioned above three different keys are used, they are:

3.2.5 Data Encryption Keys (DKs)

Keys which are generated and shared within a group and BS.

3.2.6 Re-keying Key (RK)

Key which is generated and shared between a node and BS which is used during re-keying.

3.2.7 Secret Key (SK)

Key which is shared between a node and BS. The keys DK and RK were encrypted using SK and maintained in its volatile memory. Due to this little bit of computational overhead, even if the nodes are physically captured, the keys cannot be retrieved from its volatile memory.

3.3 Assumptions and Pre-requisites to A'SDRP for Wireless Sensor Network

3.3.1 Network Assumption

A new modified approach assumes wireless sensor network in which the nodes are static with similar computational and communication capabilities. The network uses skipjack algorithm for encryption and decryption process. We have chosen this algorithm because the memory requirement is very less and encryption/decryption and key setup efficiency is also good [14]. To have variations in having the keys, we have used logical grouping of nodes for maintaining different set of keys. In a group, all the nodes maintain same set of keys, but every node

uses different key for different communications with the Base Station (BS).

3.3.2 Design Goals

The proposed approach is designed to identify the DoS attack, Packet Replay attack and Sybil attack. Identifying those attacks helps to increase the lifetime of the network.

3.3.3 Grouping of Nodes

If all the nodes in the network are using same set of keys, all the nodes have to participate in re-keying which is an overhead. To reduce this overhead, the nodes are grouped based on the size of the network. After grouping, if any one node needs re-keying, the other nodes in that group itself have to participate in re-keying process. This avoids the overhead of re-keying for the remaining nodes which belongs to other group(s). Let the number of nodes be N, types of key be NK, number of groups be NG and number of DKs NDK per node is limited to 9.

3.3.4 Node Deployment

Before a node is deployed, the static SK has to be embedded in the source code and convert the same to its executable (.exe) format and loaded in the node's non-volatile memory. Then every node is pre-distributed with 2 pairs of parameters, say (ki, ki-1) and (ri, ri-1) which are used for generating DKs and RKs respectively using one way hash function. A unique seed value Seedi is preset in every node during deployment [36].

The counter value C_i used by the key selection protocol for all the sensor nodes is initialized as:

$$\forall_{i=1}^{N} C_i = Seed_i$$

3.3.5 Key Selection Protocol

Every sensor node maintains a key pool *kp* of 9 keys, which are generated by the node immediately after its deployment. We are limiting the NDK as 9 since our key selection protocol uses a function Sum of Digits (SoD) which always results in a single digit. If NDK is increased, it increases the security, but leads to increase in computation overhead during key generation and re-keying process [36].

The keys which are generated by all the nodes of a group are same, but the selection of key for the current communication is not same. Figure 1- shows the packet format which carries the data; it includes the type of packet, destination ID, source ID and encrypted message which contain the value, source node ID and the key number *kn* which is used for current encryption [36].

Pkt_Type	*Dest ID*	*Src ID*	*Emsg*

"Figure 1. Data Packet Format"

All the nodes in the network maintaining a counter value which is initialized with a seed value during the deployment. In our example scenario, counter value of node *a*, and node *b* are initialized with $seed_a$ and $seed_b$ respectively. This counter value is incremented by 1 for each constant time interval T_{secs}. The nodes are deployed at any time interval. During the deployment of sensor nodes, the BS maintain the time interval $T_{dep,node\_num}$ at which the nodes are deployed. In this scenario, we assume that "node *a"* is deployed in *0th* time interval T*0* and "node *b"* is deployed in *2nd* time interval *2* of *BS* [36].

Data transmission between node *a* and BS occurs in different time slots. Data transmission between node b and BS occurs in same time slot. From this scenario, we prove that the key selection protocol chooses the right key when the BS receives the packet at same time slot and different time slots. Node *a* transfers a packet at time *Ta* during the time interval *3*. It computes the key *k* to encrypt the *msga* [36] using the key selection protocol as:

$$\varepsilon_{a,\tau_3} = 3 + Seed_a$$

$$x = SoD\left((ID_a >> (3 + Seed_a)) \times (3 + Seed_a)\right)$$

$$kn_a = x$$

$$k = kp[kn_a]$$

3.3.6 Re-Keying

In this proposed approach, re-keying is initiated by the sensor node only if any two (other than the last two) consecutive keys are invalidated or compromised due to the attacks launched in the network. Once all the sensor nodes are ready to deploy in the field, two parameters ri and ri-1 have to be preset. A new RK is generated by one-way hash function to communicate with the BS. This ri ri-1 pair is different for every nodes. Like the Dks the consecutive Rks were also generated using the previous keys.

The protocol for Re-keying mechanism between a node and BS is given [36] through folowing steps:

Step 1. Node _ BS: Firstly, the node which needs to re-key the existing DKs sends a request to the BS using RKRQ message.

$RKRQ, BS, SrcID, ERK_{n_i}(SrcID, GrpNo, H(SK, SrcID \oplus GrpNo))$

Step 2. BS _ Node: Secondly, the BS has to authenticate the node using RKA1 message which includes the time stamp.

$RKA1, DestNode, BS, ERK_{n_i}(DestID, H(SK, RDT), RDT, T1)$

Step 3. Node _ BS: Thirdly, after receiving the message RKA1, the sensor node generates a hash value using RDT and T1 and compares with the hash value sent by the BS.

$$RKA2, BS, SrcNode, ERKn_i (H (SK, RDT \oplus T1), T2)$$

Step 4. BS _ Node: Fourthly, the BS compares the hash value in the RKA2 message with hash value generated by it using the RDT, T1 and T2. If both are same, the BS sends the parameters for generating the DK to the sensor node using the RKPM message.

$$RKPM, DestNode, BS, ERKn_i (DestID, k_i, k_{i-1}, H (SK, k_i \oplus k_{i-1} \oplus T2), RDT, T3)$$

Step 5. Node _ BS: Finally the node which receives the parameters has to send an acknowledgment to BS using RKPA message.

$$RKPA, BS, SrcNode, ERKn_i (H (SK, RDT \oplus T3), T4)$$

4. Advanced SDRP: A'SDRP

The existing reprogramming protocols based on the centralized approach in which only the base station has the authority to initiate reprogramming. In the proposed system author introduce modified distributed reprogramming for WSNs. In this case, the network owner can also assign different reprogramming privileges to different users.

To provide secure and distributed reprogramming, a naive solution is to pre-equip each sensor node with multiple public key/ reprogramming-privilege pairs, each of which corresponds to one authorized user [36]. This solution is not applicable to WSNs, because resource constraints on sensor nodes often make it undesirable to implement such an expensive algorithm. This scheme allows a network user to sign a program image with his private key such that each sensor node can verify whether the program image originates from an authorized user. However, this solution is not applicable to WSNs due to the following facts. First, resource constraints on sensor nodes often make it undesirable to implement such an expensive algorithm [33]. There is another alternative certificate based approach (CBA) doesn't suits to WSN, because it is not efficient in communication, a large per-message overhead will result in more energy consumption on each sensor node [33].

To provide secure and distributed reprogramming, a naive solution is to pre-equip each sensor node with multiple public key/ reprogramming-privilege pairs, each of which corresponds to one authorized user. This solution is not applicable to WSNs, because resource constraints on sensor nodes often make it undesirable to implement such an expensive algorithm This scheme allows a network user to sign a program image with his private key such that each sensor node can verify whether the program image originates from an authorized user. However, this solution is not applicable to WSNs due to the following facts. First, resource constraints on sensor nodes often make it undesirable to implement such an expensive algorithm [33]. There is another alternative certificate based approach (CBA) doesn't suits to WSN, because it is not efficient in communication, a large per-message overhead will result in more energy consumption on each sensor node [33].

A more suitable approach is for each authorized user to send a new program image to the nodes through a standard group signature technique. A group signature scheme allows one member of the group to sign a message such that any verifier can verify that the message originated from a group member. Thus, only the group public key is preloaded onto each sensor node. Meanwhile, any group signature can be "opened" by the group manager (i.e., the network owner) to reveal unambiguously the identity of the actual signer. Unfortunately, a group signature algorithm does not support different levels of user authorities. That is, the network owner cannot specify a re-programming privilege for each user [36].

4.1 A'SDRP: The Protocol

Based on the design considerations, author propose a novel identity-based signature scheme for distributed reprogramming in WSNs. Through the proposed scheme, efforts on certificate management and the transmission overhead can be significantly reduced. Meanwhile, only the system public parameters are loaded on each sensor node. Compared with the traditional public-key cryptosystems, elliptic curve cryptography provides a good solution in terms of key size, computational efficiency, communication efficiency [36].

The A'SDRP consists of three phases:

1. System initialization
2. User pre processing
3. Sensor node verification.

In the *system initialization* phase, the network owner creates its public and private-keys and then assigns the reprogramming privilege and the corresponding private-key to the authorized user(s). Only the system public parameters from the network owner are loaded on each sensor node before deployment.

In this section, the network owner executes the following steps.

Phase 1: Let G be a cyclic additive group generated by P, GT be a cyclic multiplicative group, and G and GT have the same primer order q. Let $\hat{e}$: G × G → GT be a bilinear map.

Phase 2: Randomly pick a random number $s \in Z*q$ as the master key, and compute the corresponding public key $PK\text{owner} = s \cdot P$.

Phase 3: Choose two secure cryptographic hash functions $H1$ and $H2$, where

$H1: \{0, 1\}* \rightarrow G$ and $H2: \{0,1\}* \rightarrow Z*q$.

Then, the system public parameters are

$params = \{G, GT, \hat{e}, q, P, PK\text{owner}, H1, H2\}$, which are loaded in each sensor node before deployment.

Phase 4: Consider a user Uj with identity $UIDj \in \{0, 1\}*$ who registers to the network owner. After verifying his registration information, the network owner first sets Uj 's public key as

$PKj = H1(UIDjPrij) \in G$ and computes the corresponding private key $SKj = s \cdot PKj$.

Then, the network owner sends $\{PKj, SKj, Prij\}$ back to Uj using a secure channel, such as the wired Transport Layer Security protocol. Here, $Prij$ denotes the level of user privilege such as the sensor nodes set with specified identities or/and within a specific region that user Uj is allowed to reprogram, and subscription period (i.e., the beginning time and the end time).

In the *user pre-processing* phase, if a network user enters the WSN and has a new program image, it is required to construct the reprogramming packets and then send them to the sensor nodes. Assume that user Uj enters the WSN and has a new program image. Uj takes the following actions [36]:

Action 1: Uj partitions the program image to Y fixed-size pages, denoted as page 1 through page Y. Uj splits page i ($1 \le i \le Y$) into N fixed-size packets, denoted as $Pkti,1$ through $Pkti,N$. The hash value of each packet in page Y is appended to the corresponding packet in page $Y-1$. For example, the hash value of packet $PktY, 1$, $h(PktY,1)$, is included in packet $PktY-1,1$. Here, $PktY, 1$ presents the first packet of page Y. Similarly, the hash value of each packet in page $Y-1$ is included in the corresponding packet in page $Y-2$. This process continues until Uj finishes hashing all the packets in page 2 and including their hash values in the corresponding packets in page 1.

Action 2: With the private key SKj, Uj can compute the signature σj of the message m, where $\sigma j = 2(m) \cdot SKj$.

Action 3: Uj transmits to the targeted nodes the signature message $\{UIDj, Prij,m, \sigma j\}$, which serves as the notification of the new code image. A'SDRP relies on the underlying Deluge protocol to distribute packets for a given code image.

In *Sensor Node Verification*, Upon receiving a signature message $\{UIDj, Prij,m, \sigma j\}$, each sensor node verifies it as follows [36]:

1) The sensor node first pays attention to the legality of the programming privilege $Prij$ and the message m. For example, the node needs to check whether the identity of itself is included in the node identity set of $Prij$. Only if they are valid, the verification procedure goes to the next step [36].

2) Given the system public parameters $\{G, GT , \hat{e}, q, P, PK\text{owner}, H1, H2\}$ assigned by the network owner, the sensor node performs the following verification [36]:

$$\hat{e}(\sigma j, P) = \hat{e}\ (H2(m) \cdot H1(UIDjPrij), PK\text{owner}) . \quad (1)$$

If the equation holds, the signature σj is valid because

$$\begin{aligned}&\hat{e}(\sigma j, P)\\ &= \hat{e}\ (H2(m) \cdot SKj, P) = \hat{e}\ (H2(m) \cdot s \cdot PKj, P)\\ &= \hat{e}\ (H2(m) \cdot PKj, s \cdot P)\\ &= \hat{e}\ (H2(m) \cdot PKj, PK\text{owner})\\ &= \hat{e}\ (H2(m) \cdot H1(UIDjPrij), PK\text{owner}) .\end{aligned}$$

3) If the aforementioned verification passes, the sensor node believes that the message m and the privilege $Prij$ are from an authorized user with identity $UIDj$. Hence, the sensor node accepts the root of the Merkle hash tree constructed for page 0. Thus, the nodes can authenticate the hash packets in page 0 once they receive such packets, based on the security of the Merkle hash tree. The hash packets include the hash values of the data packets in page 1. Therefore, after verifying the hash packets, a node can easily verify the data packets in page 1 based on the one-way property of hash functions. Likewise, once the data packets in page i have been verified, a sensor node can easily authenticate the data packets in page $i + 1$, where $i = 1, 2, \ldots, Y-1$. Only if all verification procedures described previously pass, the sensor node accepts the code image [36].

5. Performance Evaluation

In this paper author discussed two different protocols for secure data transfer and reliable reprogramming of the sensor nodes. While in the centralized approach the base station holds the key for reprogramming, A'SDRP opens new gates by enabling a mechanism by which multiple users can reprogram, without having to involve the base station always. The centralized, dynamic key approach, has taken one step forward towards achieving reliable and secure WSN. Whereas, a A'SDRP based WSN opens up a new legacy of applications for WSN and a capability of having lesser memory on the memory of each of the sensor nodes. As a technology A'SDRP holds good for a new trend of applications and looks forward into the future, when the dynamic key approach is a one

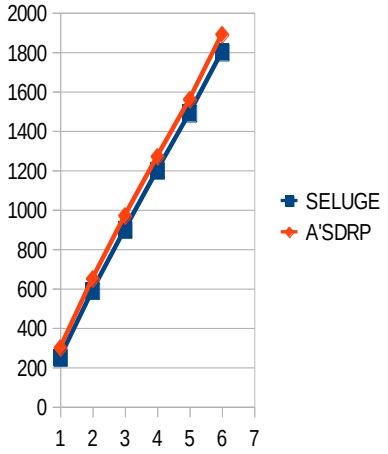
step ahead within the centralized approach solution space [36].

Both A'SDRP and SELUGE protocol are compared based on the propagation delay.

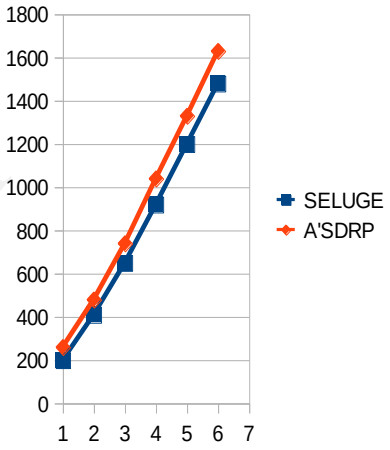
1) Average data propagation delay between two nodes (Figure 2).
2) Total propagation delay between source and destination (Figure 3), and
3) Propagation delay between two nodes (considering an overhead due to the user - Figure 4).

For the Figure 2, (see the graph) the average data propagation delay between two nodes is taken and based on the size of the code snippet the two algorithms are compared. In the simulation the propagation delay by the WSN itself it considered to be equal, but the data overhead is calculated for each of the protocols. Based on the steps needed to encrypt and decrypt additional time is considered for each of the two protocols. The Network is considered as 98% reliable (in this simulation) and data the additional delays with both these approaches are averaged out. Each data packet is considered to be 1K for simplicity sake and the above values are calculated. Based on the above assumptions for a Data size (program size) of 1KB, 2KB, 3KB, and so on, the average propagation delay is shown above. In figure 2, as the code image size increases, the propagation delays of all schemes increase almost linearly. The signature verification by sensor nodes in A'SDRP only has low impact on the propagation delay of reprogramming. For suppose , When the data size is 5 KB, the propagation delay of A'SDRP is only 3.2% more than that of SELUGE. Because, only the signature verification time of the first node has impact on the propagation delay.

In Figure 3 the total delay between two nodes (source and destination) is compared. The time required to transfer a 5KB from one node to its next in the data path is considered as 1 entry in the x- axis. So if you consider there are 5 such paths between source and the destination, A'SDRP is only 4% more than that of SELUGE. Assuming that this graph follows the same slope, a 100 nodes data hop would give the overall A'SDRP protocol a very less persentage more than to SELUGE. The signature verification time of A'SDRP takes less than 2.1% of the total dissemination time. Considering the benefits that A'SDRP provides, this time consumption is acceptable [36].

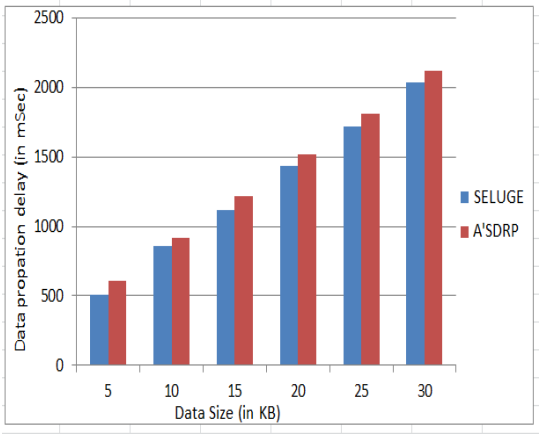


"Figure 2. Average data propagation delay between two nodes"



"Figure 3. Total propagation delay between source and destination"

In Figure 4 would suggest that the A'SDRP takes a bit longer time to transfer data but is extremely close SELUGE and DELUGU two known centralized approaches are discussed in [36]. For a 5KB data SELUGE took 512ms, A'SDRP took 604ms. So, A'SDRP with the overhead of the user taking time to get permissions takes 10-15% more time. However this is a small price to pay for reprogramming an entire WSN given the functional benefits A'SDRP has to offer. However the time gap is almost similar and with A'SDRP there are numerous other benefits like scalability and per-user privileges.



"Figure 4. Propagation delay between two nodes with user overhead"

6. Conclusion and Future Work

In the literature, a numerous of reprogramming protocols have been proposed, but these protocols did not address the distributed approach. Here author anlyse the existing SDRP protocol-key features and modify the existing one and omit the deluge protocol due to its low performance. This paper intraduce a brief survey on existing SDRP protocol with few of the modifications. In addition to the delay on WSN, reliability of the network, and encryption/decryption delays are introduced, in this step some delay is assumed for the user/server to request for reprogramming. In case of A'SDRP (modified approach) since the user sends a request and gets back the permissions from the network owner. This step takes some time and this overhead adds to the total delay. SELUGE has very minimal overhead compared to A'SDRP, as it does not need any additional permission, as there is no separate entity called user. Effectively A'SDRP takes slightly more time, compared to SELUGE. But this difference is very less and hence given the overall goodness A'SDRP adds with regards to its scalability and reliability. A'SDRP is the futuristic protocol for the security of a WSN.

Further work in this field using A'SDRP principles of WSN security during reprogramming can be extended to perform better under all phases of a WSN usage and better relative to traditional centralized approaches.

7. References

[1] V.C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches", *IEEE Trans. Ind. Electron.*, Volume 56, No. 10, pp. 4258–4265, Oct. 2009.

[2] V.C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid", *IEEE Trans. Ind. Electron.*, Volume 57, No. 10, pp. 3557–3564, Oct. 2010.

[3] Mayank Saraogi, "Security In Wireless Sensor Networks ", University of Tennessee, Knoxville.

[4] Al-Sakib Khan Pathan , Hyung-Woo Lee, Choong Seon Hong, "Security in Wireless Sensor Networks: Issues and Challenges", Proc. ICACT 2006, Volume 1, pp. 1043-1048, 20-22 Feb,2006.

[5] Culler, D.E. and Hong, W., "Wireless Sensor Networks", Communication of the ACM, Volume 47, No. 6, pp. 30-33, June 2004.

[6] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y, and Cayirci, E., "Wireless Sensor Networks: A Survey", Computer Networks, Volume 38, pp. 393-422, 2002.

[7] Kalpana Sharma, M.K. Ghose, Deepak Kumar, Raja Peeyush KumarSingh,Vikas Kumar Pandey. "A Comparative Study of Various Security Approaches Used in Wireless Sensor Networks", In IJAST, Volume 7, April 2010.

[8] Kalpana Sharma, M K Ghose. "Wireless Sensor Networks: An Overview on its Security Threats", IJCA Special Issue on "Mobile Ad-hoc Networks, 2010

[9] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, and David Culler, "Wireless Sensor Networks for Habitat Monitoring", In First ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

[10] Robert Szewczyk, Joseph Polastre, Alan Mainwaring, and David Culler, "Lessons from a Sensor Network Eexpedition", In First European Workshop on Wireless Sensor Networks (EWSN 04), January 2004.

[11] Hamid, M. A., Rashid, M-O., and Hong, C. S., "Routing Security in Sensor Network: Hello Flood Attack and Defense", to appear in IEEE ICNEWS 2006.

[12] M. Hefeeda and M. Bagheri, "Forest Fire Modeling and Early Detection using Wireless Sensor Networks", Ad Hoc & Sensor Wireless Networks, Volume 7, pp. 169–224, 2009.

[13] X. Chen, K. Makki, K. Yen, N. Pissinou, "Sensor Network Security: A Survey", IEEE Communications Surveys & Tutorials, Volume 11, No. 2,pp. 52-73,2009.

[14] Y. W. Law, J. Doumen and P. Hartel, "Survey and Benchmark of Block Ciphers for Wireless Sensor Networks", ACM Transactions on Sensor Networks, Volume 2, Issue 1, pp. 65–93, 2006.

[15] J. Chen, X. Cao, P. Cheng, Y. Xiao, and Y. Sun, "Distributed Collaborative Control for Industrial Automation with Wireless Sensor and Actuator Networks", *IEEE Trans. Ind. Electron.*, volume 57, No. 12, pp. 4219–4230, Dec. 2010.

[16] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building-Environment Control with Wireless Sensor and Actuator Networks: Centralized versus Distributed," *IEEE Trans. Ind. Electron.*, Volume 57, No. 11, pp. 3596–3605, Nov. 2010.

[17] Crossbow Technology Inc., Milpitas, CA, Mote In-Network Programming User Reference, 2003.

[18] J.W. Hui and D. Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network

Programming at Scale", in *Proceedings of ACM SenSys*, pp. 81–94, 2004.
[19] TinyOS: An open-source OS for the networked sensor regime. [Online]. Available: http://www.tinyos.net/
[20] J. Deng, R. Han, and S. Mishra, "Secure code distribution in dynamically programmable wireless sensor networks", in *Proceedings of ACM/IEEE IPSN*, pp. 292–300, 2006.
[21] P.K. Dutta, J. W. Hui, D. C. Chu, and D. E. Culler, "Securing the Deluge Network Programming System," in *Proceedings of ACM/IEEE IPSN*, pp. 326–333, 2006.
[22] C. Parra and J. Macias, "A Protocol for Secure and Energy-Aware Reprogramming in WSN," in *Proceedings of IWCMC*, pp. 292–297, 2009.
[23] N. Bui, O. Ugus, M. Dissegna, M. Rossi, and M. Zorzi, "An Integrated System for Secure Code Distribution in Wireless Sensor Networks", in *Proc. IEEE PERCOM Workshops*, pp. 575–581, 2010.
[24] S. Hyun, P. Ning, A. Liu, and W. Du, "Seluge: Secure and Dos-resistant Code Dissemination in Wireless Sensor Networks", in *Proc. ACM/IEEE IPSN*, pp. 445–456, 2008.
[25] R. Zhang, Y. Zhang, and K. Ren, "DP2AC: Distributed Privacy-Preserving Access Control in Sensor Networks," in *Proceedinsgs of IEEE INFOCOM*, pp. 1251–1259, 2009.
[26] Geoss.[Online,Available:http://www.epa.gov/geoss/
[27] NOPP.[Online, Available: http://www.nopp.org/
[28]ORION.[Online]Available: http://www.oceanleadership.org/2004/ocean-research-interactive-observatory-networks-project-office-managerselected/
[29] W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, and A. Khalili, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks", *ACM Transactions on Information System Security*, Volume 8, No. 2, pp. 228–258, May 2005.
[30] L. Gu and J. A. Stankovic, "t-kernel: Providing Reliable OS Support for Wireless Sensor Networks", in *Proceedings ACM SenSys*, pp. 1–14, 2006.
[31] D. He, L. Cui, H. Huang, and M. Ma, "Design and Verification of Enhanced Secure Localization Scheme in Wireless Sensor Networks", *IEEE Trans. Parallel Distribuded Systems.*, Volume 20, No. 7, pp. 1050–1058, Jul. 2009.
[32] D. Boneh and M. Franklin, "Identity-based Encryption from the Weil Pairing", in *Proc. Crypto*, Volume 21, No. 39, *LNCS*, pp. 213–229, 2001.
[33] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York: Springer-Verlag, 2004.
[34] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Chang, "Energy Analysis for Public Key Cryptography for Wireless Sensor Networks", in *Proceedings of IEEE PERCOM*, pp. 324–328, 2005.
[35] R. Merkle, "Protocols for Public-Key Cryptosystems", in *Proceedings of IEEE S&P*, pp. 122–133, 1980.
[36] Daojing He,Chun chen, Sammy Chan, Jiajun Bu, "SDRP: A Secure and Distributed Reprogramming Protocol for Wireless Sensor Networks", IEEE Transactions on Industrial Electronics, Volume 59, No. 11, pp. 4155-4163, Nov. 2012.