ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

Professionally Providing Security in Wireless Sensor Networks

[1]Ramesh Kammindi, [2]G.Padmarao
[1]Final M Tech Student, [2]Associate Professor
[1,2]Dept of Computer Science and Engineering
[1,2]JITS college of Engineering, Kalagampudi, -534268, w.g.dt, A.P.

Abstract: Security has become a challenge in wireless sensor networks. Low capabilities of devices, in terms of computational power and energy consumption, make difficult using traditional security protocols. Two main problems related to security protocols arise. Firstly, the overload that security protocols introduce in messages should be reduced at a minimum; every bit the sensor sends consumes energy and, consequently, reduces the life of the device. Secondly, low computational power implies that special cryptographic algorithms that require less powerful processors need to be used. The combination of both problems leads us to a situation where new approaches or solutions to security protocols need to be considered. These new approaches take into account basically two main goals: reduce the overhead that protocol imposes to messages, and provide reasonable protection while limiting use of resources. The Proposed scheme builds on the Secure Hierarchical In-Network Aggregation [2], in order to achieve not only secure but also efficient WSN data collection over a series of aggregations. We have described a basic version of our scheme, sufficient to satisfy the stated specification.

Keywords – sensor networks, location monitoring.

I. Introduction

A **wireless sensor network (WSN)** consists of spatially distributed autonomous sensors to *monitor* physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling *control* of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

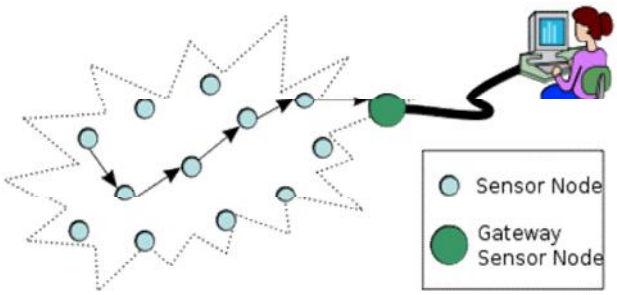

Figure 1: Wireless Sensor Network Architecture

II. Data Aggregation in the WSN

Typically, there are three types of nodes in WSN: normal sensor nodes, aggregators, and a querier.
The aggregators collect data from a subset of the network, aggregate the data using a suitable aggregation function and then transmit the aggregated result to an upper aggregator or to the querier who generates the query. The querier is entrusted with the task of processing the received sensor data and derives meaningful information reflecting the events in the target field. It can be the base station or sometimes an external user who has permission to interact with the network depending of the network architecture. Data communication between sensors, aggregators and the querier consumes a large portion of the total energy consumption of the WSN. The WSN in figure 1 contains 16 sensor nodes and uses SUM function to minimize the energy consumption by reducing the number of bits reported to the base station. Node 7, 10-16 are normal nodes that are collecting data and reporting them back to the upper nodes whereas nodes 1-6, 8, 9 are aggregators that perform sensing and aggregating at the same time. In this example 16 packets traveled within the network and only one packet is transmitted to the base station. However, the number of traveling packets would increase to 50 packets if no data aggregation exists. This number of packets has been computed for one query.

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

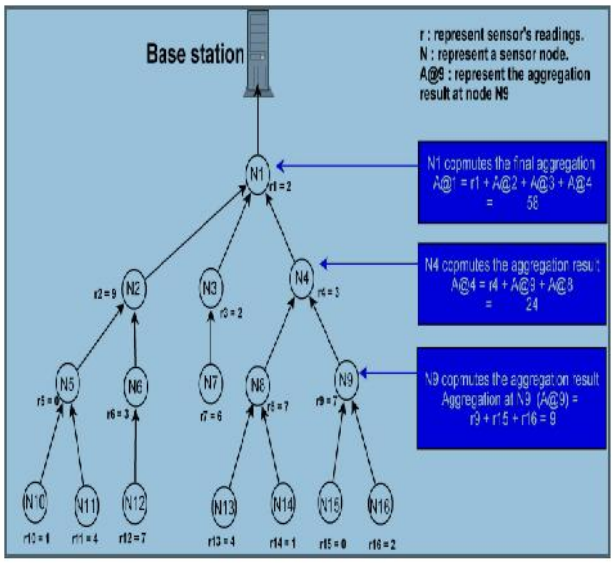

Figure 2: Secure Data Aggregation

A general definition for secure data aggregation is the efficient delivery of the summary of sensor readings that are reported to an off-site user in such a way that ensures these reported readings have not been altered (Przydate k et al. 2003). They consider an aggregation application where the querier is located outside the WSN and the base station acts as an aggregator. Moreover, a detailed definition of secure data aggregation is proposed as the process of obtaining a relative estimate of the sensor readings with the ability to detect and reject reported data that is significantly distorted by corrupted nodes or injected by malicious nodes (Shi & Perrig 2004). However, rejecting reported data that is injected by malicious nodes consumes the network resources, specifically the nodes' batteries, since each time the suspicious packet will be processed at the aggregator point. The damage caused by malicious nodes or compromised nodes should be reduced by adding a self-healing property to the network. This property helps the network in learning how to handle new threats through extensive monitoring of network events, machine learning and network behavior modeling. Consequently, it is believed that a secure data aggregation scheme for the WSN should have the following properties:

- Fair approximation of the sensor readings although a limited number of nodes are compromised.
- Ability to reduce the size of the data transmitted through the network.
- Data freshness and integrity are important and should be included in the scheme. However, the application type of the WSN affects the scheme designer's decision regarding whether to add the data confidentiality and availability or not.
- Dynamic response to attack activities by executing of a self-healing mechanism.
- Dynamic aggregator election/rotation mechanism to balance the workload at aggregators.

These properties should work together to provide accurate aggregation results securely without exhausting the network.

Requirements for Data Aggregation Security

Since WSNs share some properties with the traditional wireless networks, the data security requirements in the WSNs are similar to those in traditional networks (Perrig et al. 2002, Shi & Perrig 2004). However, there are some unique specifications that can only be found in WSNs, that require more attention during design process. In this section the required security properties to strengthen the security in aggregation schemes will be defined.

• Data Confidentiality

It **ensures** that information content is never revealed to anyone who is not authorized to receive it. It can be divided (in secure data aggregation schemes) into a hop-by-hop basis and an end-to-end basis. In the hop-by-hop basis, any aggregator point needs to decrypt the received encrypted data, apply some sort of aggregation function, encrypt the aggregated data, and send it to the upper aggregator point. This kind of confidentiality implementation is not practical for the WSN since it requires extra computation. On the other basis, the aggregator does not need to decrypt and encrypt data and instead of this, it needs to apply the aggregation functions directly on the encrypted data by using homomorphic encryption (Westhoff et al. 2006).

• Data Integrity

It ensures that the content of a message has not been altered, either maliciously or by accident, during transmission process. Confidentiality itself is not enough since an adversary is still able to change the data although it knows nothing about it. Suppose a secure data aggregation scheme focuses only on data confidentiality. An adversary near the aggregator point will be able to change the aggregated result sent to the base station by adding some fragments or manipulating the packet's content without detection. Moreover, even without the existence of an adversary, data might be damaged or lost due to the wireless environment.

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

• Data Freshness

It ensures that the data are recent and that no old messages have been replayed to protect data aggregation schemes against replay attacks. In this kind of attack, it is not enough that these schemes only focus on data confidentiality and integrity because a passive adversary is able to listen to even encrypted messages transmitted between sensor nodes can replay them later on and disrupt the data aggregation results. More importantly when the adversary can replay the distributed shared key and mislead the sensor about the current key.

• Data Availability

It ensures that the network is alive and that data are accessible. It is highly recommended in the presence of compromised nodes to achieve network degradation by eliminating these bad nodes. Once an attacker gets into the WSN by compromising a node, the attack will affect the network services and data availability especially in those parts of the network where the attack has been launched. Moreover, the data aggregation security requirements should be carefully implemented to avoid extra energy consumption. If no more energy is left, the data will no longer be available. When the adversary is getting stronger, it is necessary that a secure data aggregation scheme contains some of the following mechanisms to ensure reasonable level of data availability in the network:

– **Self-healing that** can diagnose, and react to the attacker's activities especially when he gets into the network and then start corrective actions based on defined policies to recover the network or a node.
– **Aggregator rotation** that rotates the aggregation duties between honest nodes to balance the energy consumption in WSN.

• Authentication

There are two types of authentication; entity authentication, and data authentication. Entity authentication allows the receiver to verify if the message is sent by the claimed sender or not. Therefore, by applying authentication in the WSNs, an adversary will not be able to participate and inject data into the network unless it has valid authentication keys. On the other hand, data authentication guarantees that the reported data is the same as the original one. In a secure data aggregation, both entity and data authentication are important since entity authentication ensures that some exchanged data between sensors. For instance, electing an aggregator point or reporting invalid aggregated results are authenticated using their identity while data authentication ensures that raw data are received at the aggregators at the same time as they are being sensed.

• Non-repudiation

Ensures that a transferred packet has been sent and received by the person claiming to have sent and received the packet. In secure aggregation schemes, once the aggregator sends the aggregation results, it should not be able to deny sending them. This gives the base station the opportunity to determine what causes the changes in the aggregation results.

• Data Accuracy

One major outcome of any aggregation scheme is to provide an aggregated data as accurately as possible since it is worth nothing to reduce the number of bits in the aggregated data but with very low data accuracy. A trade-off between data accuracy and aggregated data size should be considered at the design stage because higher accuracy requires sending more bits and thus needs more power.

III. Proposed System

We are interested in protocols that can perform a sequence of aggregations $\{A_1, A_2, \ldots, A_j\}$ in a WSN with n_a faulty nodes, and satisfy the following properties:

Security

- At most na aggregations fail.
- For every successful aggregation A ▫ {A1,A2, . . .Aj}, with VA being a multi-set of values contributed by correct nodes in the aggregation tree TA, and V ′ A a multi-set of arbitrary values in range M equal in size to the number of faulty nodes in TA, it holds that

$$val_A = agr(V_A + V'_A)$$

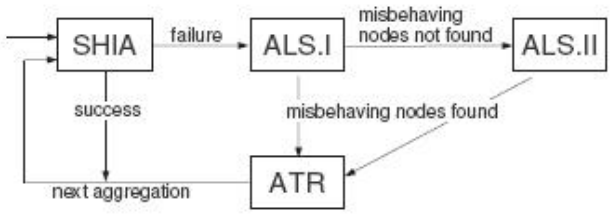

Figure 3 : System Architechture

The system operates in three stages. First, data aggregation and manipulation detection are performed by SHIA [2], due to its efficiency and effectiveness. If

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

successful, the system proceeds to the next aggregation. Otherwise, at a second stage, the Adversary Localizer Scheme (ALS) is launched: the ALS.I phase localizes, i.e., marks, nodes that disrupted the aggregation value, and ALS.II marks nodes that disrupted the acknowledgement collection during stage one (SHIA). At the third stage, the Aggregation Tree Reconstruction (ATR) protocol is invoked, which constructs a new aggregation tree excluding the marked nodes. Gradually, after a series of failed aggregations, all the faulty nodes will be excluded, allowing undisrupted in-network aggregation and thus efficient operation.

We use the following cryptographic primitives: H, a collision-resistant hash function, and MAC, a Message Authentication Code. AuthK(m) denotes message m authenticated using the symmetric key K, e.g. <m,MAC(m)K>.

Stage One: SHIA

The SHIA algorithm focuses on the sum aggregate and consists of three phases: query dissemination, aggregate commit and result checking. The base station (BS) initiates the aggregation, generating a nonce N that identifies the aggregation session and broadcasting it to the network, as part of a query (along with other possibly useful data) in an authenticated manner.

The aggregate-commit phase, every node calculates a label, based on the labels of its children and its own value, and sends it to its parent node. The label is a <count, value, commitment> tuple, with count the number of nodes in the subtree rooted at the node, value the sum of all the nodes values in the subtree, and commitment the cryptographic commitment tree over the data values and the aggregation process in the subtree.

In the result-checking phase, the BS disseminates, using an authenticated broadcast, N and the <c, v, h> label. Every node uses this label to verify if its value was aggregated correctly.

Fact 1: If a node and its child both follow the SHIA protocol, either they both acknowledge or neither do.

A node s acknowledges by releasing an authentication code (ack): MACKs (NkOK), where OK is a unique message identifier and Ks is the key shared between node s and the BS. Leaf nodes send their ack while intermediate nodes wait for acks from all their children, compute the XOR of those acks with their own ack, and forward the resultant aggregated ack to their parent.

Once the BS has received the aggregated ack message Ab from b, it can verify whether all nodes acknowledged the aggregation value: It calculates the *ack* of every sensor (using the key shared with the node), XOR'es them and compares the result to Ab. In case of equality, all nodes acknowledged, and the BS declares the aggregation successful. Otherwise, our ALS protocol is triggered.

Stage Two: Adversary Localizer Scheme, Part I

ALS.I marks nodes that misbehaved in the *aggregate commit* phase of SHIA or the dissemination of *off-path* values. ALS.I consists of two phases:

Hierarchical Collection of Confirmations: The BS initiates this phase by sending an authenticated broadcast containing N and informing all nodes that ALS.I is taking place. If a node had not acknowledged the result (as determined by SHIA), it does not respond. Otherwise, a leaf node s sends a confirmation Ms = AuthKs(N) to its parent. An internal node t waits for its children, u1, u2, ..., uk (the order is based on their identifiers) to send their confirmations. Then, t sends up the following confirmation: Mt = AuthKt(N,Mu1 ,Mu2 , ...,Muk). If t does not receive any messages from its rth child, Mur is replaced with a predefined message Mnr, indicating "no message received from this child".

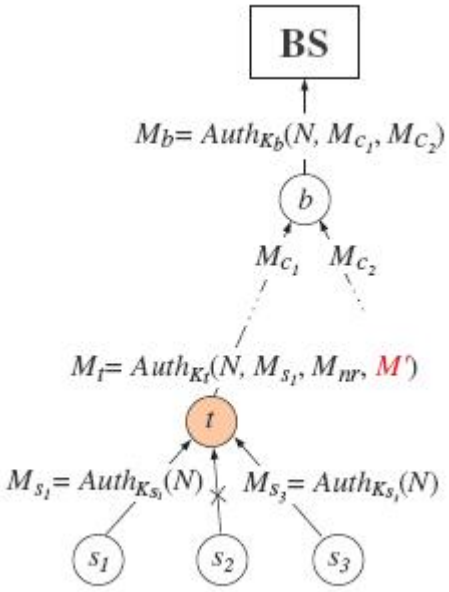

Figure 4 ALS.I: Hierarchical Collection of Confirmations

Recursive Processing of Confirmations: The above procedure results in message Mb reaching the BS. If no message is received, then b, the single child of the BS, is marked. Otherwise, the message is processed in a recursive manner. As the aggregation tree is known to the BS, it knows that Mb should be authenticated using Kb. If it is, and the message begins with N, and the proper number of *child* messages (equal to the number of b's

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

children) can be extracted from it, the message is regarded as *legitimate*, and the recursive procedure is applied to each child message. Otherwise the message is regarded as *illegitimate*. Note that the special Mnr message is also regarded as illegitimate. In that case, the BS marks the node to which this message corresponds to and its parent (in the farther recursive executions, when the BS is not the parent); the recursive execution stops. The recursive procedure also stops when it reaches a leaf node.

Stage Two: Adversary Localizer Scheme, Part II

It is possible that ALS.I does not localize any faulty nodes, even though SHIA declared a failed aggregation. This can happen when all correct nodes acknowledge, which implies a correctly done aggregation but a faulty node disrupting the aggregation of *ack*s. On the positive side, in such a situation the BS can be sure of a correct aggregation result. However, the not removed faulty node could disrupt a subsequent aggregation, something unacceptable according to our problem statement.

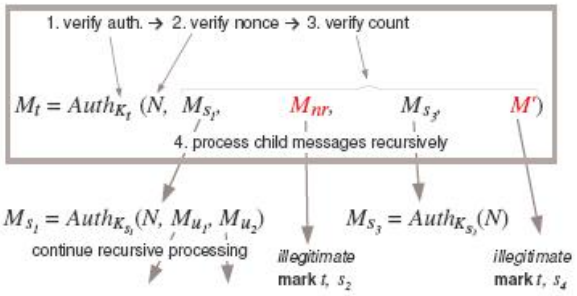

Figure 5:ALS.I: Recursive Processing of Confirmations

The ALS.II scheme addresses this problem. To implement ALS.II, we need to slightly modify SHIA, making every node store the *ack* messages it received from its children. ALS.II delivers them to the BS, using the same mechanism as the Hierarchical Collection of Confirmations in ALS.I. Then, the BS identifies inconsistencies, through an algorithm based on Recursive Processing of Confirmations.

Hierarchical Collection of acks

The BS initiates this phase by broadcasting an authenticated message containing N, informing the nodes that ALS.II is taking place. A leaf node s does not send anything. An internal node t, with non-leaf children u1, u2, ..., uq, sends up Mt = AuthKu(N,Mu1 ,Mu2 , ...,Muq ,Au1 , ...,Auq), where Mu1 ,Mu2 , ...,Muq are the messages t received from its children in this phase, and Au1 ,Au2 , ...,Auq are *ack* messages that it received from them in the SHIA result-check phase.

Recursive Processing and Ack Analysis

Upon receiving Mb from its child b, the BS recursively processes it to find the source(s) of discrepancy. As in ALS.I, if the message of node t illegitimate, meaning that it is not authenticated with Kb, it does not begin with N, or the proper number (of nonleaf children) of child messages and *ack* messages cannot be extracted from it, both t and its parent are marked. There are only two significant differences from ALS.I. First, if for some node u the *ack* message Au equals Au, then the corresponding child message Mu is not further processed. Second, the BS is looking for *ack inconsistencies*, which have two variations: (i) for node t which has a leaf node s as a child, As is different from MACKs (NkOK), the *ack* of node s. (ii) for node t which has a child s, which has the children u1, ..., uq, the value As is not equal to MACKs (NkOK) (NAui) (Fig. 4b). If an ack inconsistency is detected, both t and s are marked, but the recursive procedure is continued (if possible).

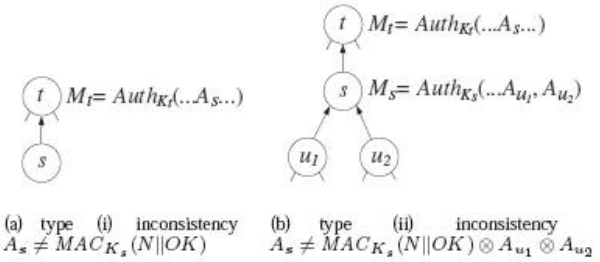

Figure 6: ALS.II: ack inconsistencies

Stage Three: Aggregation Tree Reconstruction

The Aggregation Tree (Re) Construction (ATR) protocol, in addition to the tree construction, allows the BS to exclude nodes in BL, a *black list*, from the new tree T′A and provides the BS with the knowledge of T'A. The primary *requirement* for ATR is that its output, T'A, is identical to the parts of the tree nodes know. This way, a faulty node is not be able to mislead the ALS protocols marking a correct node.

The BS initiates ATR, via a neighboring node b, by sending a *tree establishment (TE)* message <N,BL, n>, protected by a network-wide broadcast authenticator AuthBcast.

As the *TE* message is flooded, it is authenticated in a hop by hop manner by data-link broadcast authentication. Each v maintains s from which it first receives a fresh *TE* as its parent, and confirms to s that it is its child. After s hears from its k children, namely v1, . . . , vk, it sends its *response to the BS*: AuthKs (N, s, (v1, . . . , vk)), or AuthKs (N, s), if it is a leaf. The responses are

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

propagated upwards to the BS. After sending its own response, the node acts as a relay for the responses of its children and up to n responses or until the response collection concludes; these constraints are added to keep the cost bounded, but might result in loss of legitimate responses. A faulty node cannot create any inconsistency between the tree at BS and the nodes if responses are lost or dropped. Even if the faulty node eliminated a subtree, it would at most prevent aggregation from a part of the network, but no correct node would be blacklisted and thus permanently excluded.

Highly Resilient ATR

To ensure that the new tree ATR covers all nodes, we sketch here a different protocol, whose initial phase must run before any aggregation takes place. After each node s ran a secure neighbor discovery, it floods its *neighbor list (*NLs*)* across the network; a fresh NLs is relayed by each node only once. Upon receipt of the neighbor lists from all nodes, the BS constructs the network connectivity graph, rejecting links not announced by both neighbors. The BS then calculates locally TA. At the end of this initial phase, as well as after any subsequent reconstruction, the BS simply distributes the newly calculated aggregation tree across the network. It suffices that each node relays the message containing TA once at most.

We emphasize that the costly *NL* collection is performed in general *only once*, at the initial phase of ATR. To ensure resilience to DoS attacks, nodes need to authenticate each *NL* they relay. Otherwise, faulty nodes could flood the network with bogus neighbor lists. To achieve this, public key cryptography is needed; recent implementations and references within, attest to its feasibility for WSNs. Each responding s signs its NLs. The scheme cannot be exploited by clogging/energy consumption DoS attacks: Correct nodes immediately ignore messages coming from a neighbor that forwarded one invalid-signed *NL*, as the forwarder should have checked its validity already.

IV.Conclusion

Our scheme builds on the Secure Hierarchical In-Network Aggregation [2], in order to achieve not only secure but also efficient WSN data collection over a series of aggregations. We have described a basic version of our scheme, sufficient to satisfy the stated specification. However, there are a number of enhancements and extensions that could be integrated in the proposed system. For example, our scheme could interoperate the improved SHIA approach, yielding a more efficient, O(log2 n), successful aggregation

References

[1] Castelluccia, C., Mykletun, E. & Tsudik, G. (2005), Efficient Aggregation of Encrypted Data in Wireless Sensor Networks., in ‘MobiQuitous’, IEEE Computer Society, pp. 109–117.

[2] Chan, H., Perrig, A. & Song, D. (2006), Secure hierarchical in-network aggregation in sensor networks., in A. Juels, R. N. Wright & S. D. C. di Vimercati, eds, ‘ACM Conference on Computer and Communications Security’, ACM, pp. 278–287.

[3] Corporation, C. (2006), ‘Mica2 datasheet’. Reviewed 10th of October 2007, http://www.xbow.com/Products/Product_pdf_files/ Wireless_pdf/MICA2_Datasheet.pdf.

[4] Domingo-Ferrer, J. (2002), A provably secure additive and multiplicative privacy homomorphism., in A. H. Chan & V. D. Gligor, eds, ‘ISC’, Vol. 2433 of Lecture Notes in Computer Science, Springer, pp. 471–483.

[5] Du, W., Deng, J., Han, Y. S. & Varshney, P. (2003), A witness-based approach for data fusion assurance in wireless sensor networks, in ‘IEEE Global Communications Conference (GLOBECOM)’, Vol. 3, pp. 1435– 1439.

[6] Grigoriev, D. & Ponomarenko, I. V. (2003), ‘Homomorphic public-key cryptosystems over groups and rings’, CoRR cs.CR/0309010.

[7] Guimar˜aes, G., Souto, E., Sadok, D. F. H. & Kelner, J. (2005), Evaluation of security mechanisms in wireless sensor networks., in ‘ICW/ICHSN/ICMCS/SENET’, IEEE Computer Society, pp. 428–433.

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

RAMESH KAMMINDI received his B.TECH degree in IT from Jogaiah Institute Of Technology And Sciences Kalagampudi West Godavari (Dt), in 2011,and He is presently pursuing the M.TECH, Degree in CSE from Jogaiah Institute Of Technology And Sciences Kalagampudi .West Godavari (Dt). At present, He is engaged in**"Professionally Providing Security in Wireless Sensor Networks".**

G.PADMARAO received the M.Tech degree from JNTUH, SIT, Jawaharlal Nehru Technological University, Hyderabad in 2011.Currently he is working as Associate Professor in JITS Engineering College, Kalagampudi.He has twelve years of experience in teaching. Previously he has worked with KGRL college, Bhimavaram.His research interests include object oriented programming, cloud computing and data mining and data ware housing.