Secure Index on Distributed Data: UML Extension for Wireless Sensor Networks

VandanaBhasin

Abstract:***The research on wireless sensor network has evolved with applications being developed in several domains. However, applications designed for wireless sensor network are attributed with programming which occurs at lower levels of abstractions of the operating system. The application designer has to be aware of both the domain of the application and its corresponding hardware platform. This creates a strong coupling between the implemented code and hardware platform. Hence all applications are designed for specific platforms and become difficult to maintain, modify and reuse. Our proposal creates UML models for a WSN application to formalize it according to a development life cycle. Secure Index on Distributed data (SIDD) is an application that addresses two issues of data distribution and security. The UML model helps to harmonize the domains of security and application. Hence the paper focusses on converging the designing process into a formal development mode using UML representations***.

Index Terms*: *Distributed data,Security, UML Modelling, Wireless sensor network.

I. INTRODUCTION

Recently, many technological innovations in the hardware have directed the computing world to the arena of ubiquitous computing. This new paradigm has the wireless sensor network at its core. These networks have highly resource constrained nodes that have the capability to sense the physical world phenomena and have enough resources to connect with the computing world. They have beendeployed for large number of applications like protection of civil infrastructure, habitat monitoring, toxic gas detection, supply chain management and health care. These networks have numerous challenges such as a broadcast communication channel, restricted power resources and limited storage capability. Hence, these networks need to be secured for deploying them in critical areas like military surveillance and health care.

The authors in [1] had created an algorithm to secure the data of a node from being tampered. To protect data of a sensor node from being tampered, a secure index was created on distributed data. The algorithm secure index for distributed data (SIDD) has been designed for this very purpose[1]. It partitions data collected on a node into multiple chunks and distributes these chunks randomly on the network. This algorithm was devised for a heterogeneous clustered network. These chunks of data before being distributed on the network are indexed and encrypted.

Revised Manuscript Received on July 22, 2019.

VandanaBhasin, LalBahadurShastri Institute of Management, Dwarka, Delhi, India.

The partitions are stored in the network on different nodes whose address is specified by the roots of polynomial equation [2]. The scheme established end to end confidentiality as data is not aggregated at the cluster head and does not to be decrypted.

These partitions of data are accessible to the base station via the cluster head. The cluster head node is a resource rich node and has an index stored on it. The index can provide the immediate location of the node without traversing throughout the network. The indexing scheme works on each unique data item for generating the index. Hence it provides single value granularity. This scheme focusses on the implementation phase to check the feasibility of the new design. It does not consider any software development methodology to develop the design. The implementation phase involves complex software and hardware interactions of the components on the sensor node and hence, these designs are prone to errors. With the addition of security, this complexity is further enhanced. Further details of the paper can be referenced from [1].

As can be perceived, there are numerous challenges encountered while designing a WSN application [3]as these applications only consider the coding or implementation layer of a software development life-cycle model. Primarily, the emphasis remains on coding, hereby making such applications susceptible to errors. Debugging is equivalently challenging as these nodes have a restricted input output capability for displaying errors [4].Lastly, the nodes once deployed on the fields cannot be monitored for knowing whether the application is running correctly or not. Hence, I have applied UML to model the applications of wireless sensor networks, forming abstraction levels to produce a reliable application. Early analysis of the application enables the design to be evaluated before deploying the application on the actual nodes.

In this paper, I model SIDD using UML which is a de-facto standard for object-oriented software engineering. With this the external and internal architecture of the application is captured. Earlier the simulations were performed in MATLAB. Using UML representations which formalize the development process, the design of SIDD is further elucidated creating reusable, maintainable and robust design. The algorithm has not been tested on actual motes, but before being deployed on them, the design can be re-verified. The rest of the paper is organized into following sections: section 2 contains related work, section 3 discusses the UML representations for SIDD. Section 4 explains conclusion and future work.

II. RELATED WORK

UML is an object-oriented modelling language with rich annotations which are applicable in multiple domains for documentation and specification [5]. The set of building blocks, that is the UML diagrams, model the common mechanisms of any software application. The visual representations create a higher level of architecture for the application and numerous errors can be handled before the actual deployment of the application. UML has provisions for constructing different views of a software system and supports both functional and dynamic modelling[6]. The diagrams are structured in two different views:

- Static view: This view exhibits a robust structure for an application. It includes the class diagram, deployment diagram and the component diagram.
- Dynamic view: This view emphasizes on the behavioral features. The diagrams that are part of this are: use case diagrams, activity diagrams, interaction diagrams and state chart diagrams.

UML has proved its countenance in numerous domains like object tracking in videos[7], in robot design [8] and cyber physical systems [9]. There has been a lot of research around wireless sensor network applications being designed in UML. We have seen that in one of the papers, the authors have captured the features of hardware and middleware modules of WSN using design patterns of UML. They create a higher level of abstraction to represent time hours, events and application flow through state chart diagrams[10].

The authors in this paper [3] have devised a mechanism to create three levels of abstractions and automatically transform the models into final application code. However, they have considered a model driven approach to transform from meta models to final code. They have applied it to the NesC language.

In this paper [11], the authors have used activity diagram as a tool to program networks that constitute Sun SPOTS as their components. They use activity diagrams to generate executable code by using an interpreter.

In this paper [12], the authors have revisited the security attack models and analyzed them using the unified modelling language. They have used the interaction diagrams of UML to provide an in-depth knowledge of the attack at four different layers. They have facilitated the understanding of attacks at the physical, link, network and transport layer.

The authors in [13] have focused on modelling security attacks in WSN at the MAC layer. They study the behavioral patterns through sequence diagram approach of UML. They have also been able to identify a new type of attack, the explicit contention notification for hybrid MAC mechanisms. With this attack, the adversary sending false messages of explicit contention notification would increase the draining of energy and delays of the network. The authors in [14] have tried to bridge the gap between application domain experts and network experts who do not have knowledge of each other's domain. They have presented a model driven architecture to facilitate development of WSN applications. They have promoted reuse of software artefacts as they have partitioned the responsibilities between different experts. They have quantified TinyOS using model editors from GenModel.

The authors Uke et al [15]have focused on the modelling of WSN data aggregation using UML as the formal software designing tool. They have worked to generate various UML diagrams pertaining to data aggregation in WSN.

As can be seen from the literature, UML has been used in wireless sensor networks through a subset of its diagrams. They have addressed their specific problem and have transformed the models into code that can be directly deployed on nodes containing TinyOS. However, in this paper I have worked upon to amalgamate security paradigms and data distribution through UML diagrams using the formalized life-cycle model. The focus of this paper is on converging the designing process into a formal development mode using UML representations. In the next section, I discuss SIDD along with the UML representations.

III. WSN UML MODELLING NOTATIONS

In this section, I apply UML which is a de facto standard for object-oriented modelling. It provides graphical, modelling techniques which allow a user to depict different characteristics of the system. There are 13 UML diagrams [16] that support different characteristics of the system which might be either static or behavioral. These diagrams, however, are indistinguishable for certain domains; therefore, UML provides extensions. It defines stereotypes, constraints, values and tags to accommodate the ambiguity in certain domains. With these extension notations, the designers can compose models that adhere to particular application domain area.

UML is designed for an effortless transition from elaboration phase to the construction phase supporting both forward and reverse engineering. The strength of UML is that it allows real-world entities to be modelled as objects which collaborate with one another. These objects are characterized through their attributes, operations, states and constraints.

The UML diagrams depict different perspectives of a defined problem. While gathering requirements, use case diagrams are extensively used in the market as it is very easy for the developers to understand the problem perspective of end users. Activity diagrams model the sequence of activities that form a use case. They represent the flow of events. A class diagram gives the static view of objects and the relationships which is part of the analysis phase in the SDLC. The sequence diagrams, on the other hand, depict the dynamic structure of a use case which is accomplished through the interaction of objects. A component diagram depicts the components or libraries used by a node that is they give a static view of where and how each component is going to work on a node. The deployment diagram defines the interaction of the hardware and the software components of the node. These diagrams form the part of individual phases of life-cycle models and also represent an iterative way of building a project. So, depending upon the requirements of WSN application, an application might use some or all of the diagrams [17].

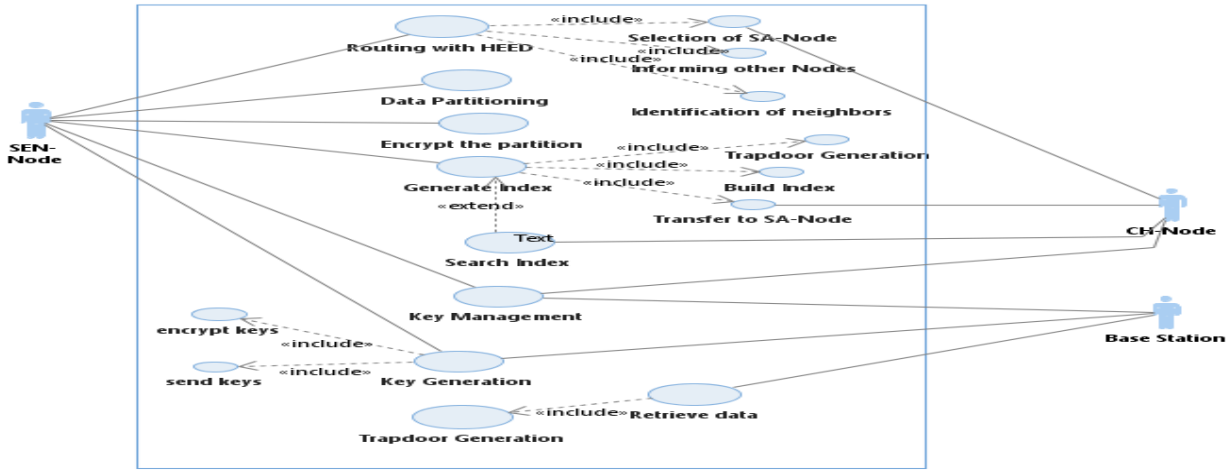

Fig 1.1: Use case Diagram for SIDD

A. Use Case Diagram

The use case diagram represents the fundamental diagram in the UML giving the functional requirements of the system and showing the interaction between the actor and the system. In case of a wireless sensor network, the interactions happen between the sensor node, the cluster node and the base station. These nodes represent the hardware that they have been deployed on; but are also initiators of events in the WSN domain. Hence, these can be annotated as actors of the system which initiate events and collect data. In this manner, we can formalize the use case diagram for SIDD as shown in fig 1.1.

The use case diagram depicts the functionalities of the application, SIDD. As can be reflected from the diagram, there are numerous functions performed by SEN-Node as key generation, key management, selection of cluster head, encrypt the partition, and generate index. To further elaborate use cases, they can be expressed using three representations. These representations provide a great potential of understanding the flow of events at the early stage of design conception and evidently help identify any missing fragments of the algorithm. The three representations for use cases are:

- The use case description
- Activity Diagram
- Tabular Description

The tabular and use case description describe have standard templates to describe the functionality of a use case. They contain header items like the use case name, the external actors, pre-conditions and post conditions. The basic flow and the exception conditions specify the sequencing of the use case and any exceptions to the basic flow of activities respectively[18]. In the fig 1.2 and 1.3, the textual representation of a use case 'Key Generation' and 'Key Management' is shown.

Use case: Key generation
Actor: Base station
Scenario: Generate keys and distribute it to the network
Precondition: An algorithm to generate keys must be selected and available on the base station.
Post condition: Keys are encrypted and copied to the sensor nodes during the pre-deployment phase
Basic flow:

1. Base station has a large set of keys, Pool, that are random and have been assigned global identification numbers.
2. Global identification number combines the node identification number with the local key number.
3. These keys are encrypted using the Vernam cipher, VR.
4. The encrypted version of the key, $Key \oplus VR$ is stored in the base station with its global identifier.
5. Subset of i keys are loaded from the $|Pool|$ with their corresponding identifiers into the sensor node.

Fig 1.2: Use case description for 'Key Generation'

Use case: Key Management
Actor: Base station, SEN-Node/Sensor Node, CH-Node/Cluster head
Scenario: Sharing of keys between each member node, cluster head and base station
Precondition: a cluster has been formed and the process of selection of cluster head is complete.
Post condition: Keys have been securely distributed
Basic flow:

1. The key management use case starts with broadcast message to the cluster head and to other SEN-Nodes in the network.
2. The cluster head sends its ID and list of neighbors to the base station.
3. Amongst a set of keys allocated to the cluster head; one key is selected for communication with other nodes Key_{CH}.
4. This key is broadcasted by the base station as $Key_{CH} \oplus VR \oplus Key_{No} \oplus VR$ to the cluster head.
5. The cluster head receives this key as $Key_{CH} \oplus Key_{No}$. The cluster head already possesses its own key, Key_{CH}. When it XoRs its key with the key received from the base station the cluster head key is cancelled and the keys for communicating with the node is obtained in a secret manner.
6. Similarly, the sensor node receives the key of the cluster head as $Key_{CH} \oplus Key_{No} \oplus Key_{No} = Key_{CH}$.
7. Rest of the 'f' keys are generated using HMAC-SHA1 algorithm at the node itself. These are encrypted using the cluster head key and sent to the base station.
8. Hence the total of $i + f = r$ keys are used for key management use case.

Fig 1.3: Use case description for 'Key Management'

The use case descriptions have distinctly stated the objective of the problem domain. The pictorial representation gives a clear insight into the problem domain both to developer and the end user.

With this formalization of the design process, it was found that in the SIDD algorithm, the process of exchanging neighbor information had been duplicated once in the clustering algorithm and also during the key management process. This duplication can be verified and handled appropriately. The formalized design process has helped to identify duplication of information.

This would certainly optimize thealgorithm resulting in improved performance and reliability as is the case for any software development life-cycle model.

B. Activity Diagram

Another way to represent a use case functionality is through an activity diagram. It defines the sequencing of temporal events. We have depicted the data partitioning mechanism through the notations of an activity diagram as shown in figure 1.4.

The data partitioning scheme starts with the collection of data. Since data collected needs to be partitioned into chunks, the chunk size can be fixed as has been already done in SIDD.But for optimizing the algorithm, it can also be varied by passing as a parameter. The scope of the existing algorithm can be improved. The SEN-Node then selects 'n' -1 random numbers from the finite field defined by prime numbers, the upper limit is definedby identifying the total number of nodes in the network. Then the node needs to calculate the n^{th} root of the polynomial equation as specified in the action state of the activity diagram in fig 1.4. This process is followed by indexing and encrypting of the partition. Then, one of the random roots is selected and the node sends the encrypted partition to the specified address.

The activity diagram emulates a flowchart but has more constructs like swim lanes which apply when there are more actors involved in the use case. It allows for object flows that are states of an object during the use case being performed.

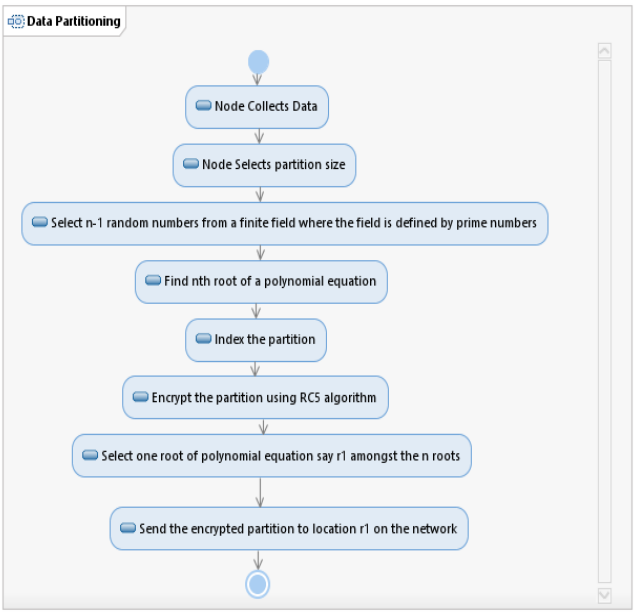

Figure 1-4 Activity Diagram for Data Partitioning

As can be seen that UML models provide high level of abstraction to visually create, design and maintain the application.In figure 1.4, activity 'Index the partition' is an action state for data partitioning.But,it can be furtherdecomposedinto more action statesfurnishing more details about it.

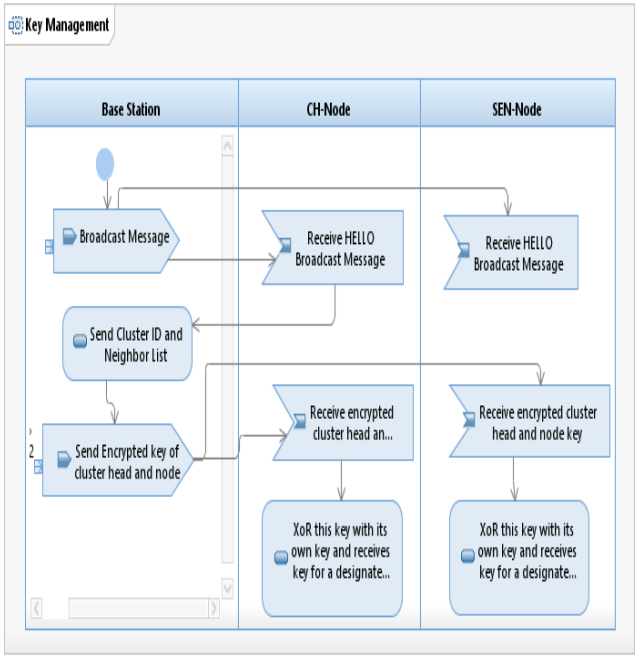

Figure 1-5 Activity Diagram for Key Management

The activity diagram for 'Key Management' use case is shown in fig 1.5. This is a use case where all three actors are involved to complete the entire activity.Hence, it illustrates the swim lanes for the three different actors alreadyclassified.In activity diagram, action states depict the processing activities at the node. But this diagram,has provisions for showing the signals or messages broadcasted by the base station. Hence the symbols for signals are shown in the swim lane for Base Station, CH-Node and SEN-Node. The Base station uses the sending signal symbol and CH-Node and SEN-Node the receiving signal symbol. The steps shown diagrammatically have been discussed in fig 1.3 also. In the next section, the sequence diagrams are discussed.

C. Sequence Diagram

Sequence diagrams are the dynamic diagrams in UML which depict the time-based events for the objects in the problem domain. These diagrams show the interactions for the objects on a timeline. However, in the scenario for SIDD the sequence diagrams are considered for two use cases: key management and for the entire system.

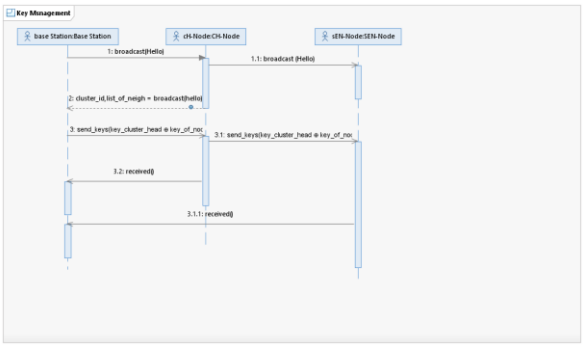

Figure 1-6 Sequence Diagram for Key Management

The key exchange mechanism has three objects 'baseStation' associated with the class Base Station and similarly objects for cluster head, 'cH-Node' and 'sEN-Node'. These objects interact with each other sending messages and reflect the behavioral characteristics of the use case.

In the fig 1.7, the sequence diagram for the system is drawn to show the sequence of messages on the timeline. The timeline indicates that the messages traversed are from the top to the bottom.Thediagram depicts that the algorithm starts with generation of keys that are deployed on all nodes irrespective of them being a cluster head or a sensor node. All the nodes also generate a subset of keys using HMAC-SHA1 algorithm. After completing the key generation and collection of network statistics for routing, the nodes start sensing the data. The nodes also generate roots from a finite field and calculate a final root to generate a polynomial equation. The number of partitions of data are decided by the number of roots as the roots specify the addresses of the nodes where the slices will be routed.

The sensed data is now given to the trapdoor generation module for index creation. Index is created for a slice of data using build_index. The index is then communicated to the CH-Node. In the meanwhile, the SEN-Node or sensor encrypts that slice of data and sends it to the respective location on the network.

The base station retrieves data from the nodes through the cluster head. It must initially generate a trapdoor for the item it is searching for. The trapdoor is sent to the cluster head which searches the indices for the data item. When the cluster head responds with an affirmation for the data item to the base station, the base station sends a data retrieval message to the network and data is received by the base station.

In the sequence diagram for the entire system, the interactions between the objects are minimal as these interactions imply sending messages over the communication channel.

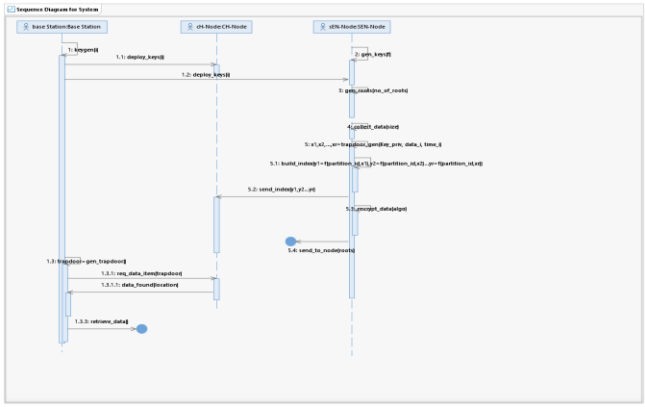

Figure 1-7 Sequence Diagram for entire system

In the wireless sensor networks, the energy consumption can be minimized only when communication over the channel is the least. Hence, the diagram elucidates it as most functions are executing on the timeline of the same node. Hence this diagram co-relates with situation of a WSN.

D. Communication/Collaboration Diagram

These diagrams depict another way of representing the flow of messages. They also depict the dynamic behavior of the use case. They represent an extremely powerful way to depict coupling. Fig 1.8 and 1.9 show the communication diagram for 'Key Management' use case and the entire system respectively.

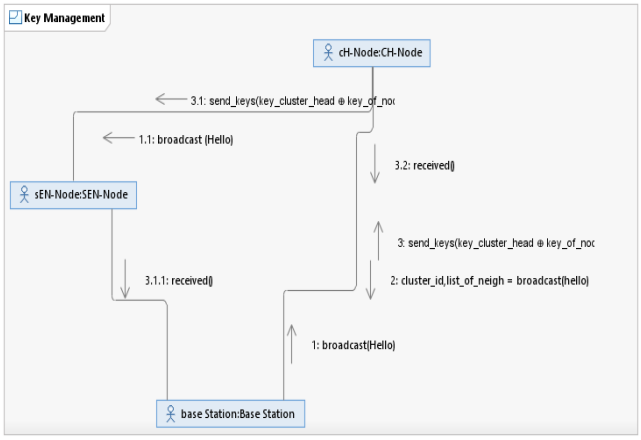

Figure 1-8 Communication Diagram for Key Management

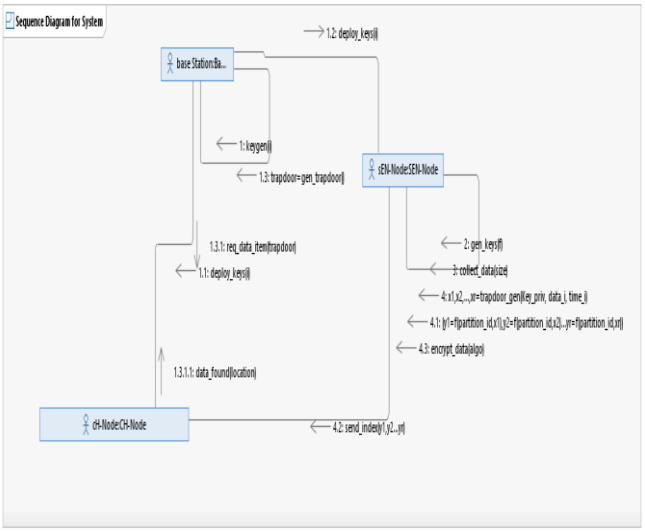

Figure 1-9 Communication Diagram for Entire system

E. Class Diagram

UML Class diagrams allow for structural representation of problem domain objects. The classes are organized in a static way through associations between them. These diagrams closely relate to the coding solutions as they can be used directly to generate code in any object-oriented programming language. A class denotes instances of a set of objects. Graphically, a class in UML is rendered as a rectangle with three compartments.

Each of the compartment depicts the class name, its attributes followed by its operations. The association between the classes indicate that these classes will be interacting with each other in different scenarios. There are specialized associations such as aggregation and generalization. Aggregation is a whole-part relationship between classes indicated by either by a filled or empty diamond. The generalization relationship between two classes is a parent/child relationship and it is symbolized using a triangle in front of a line [19].

The class diagramin fig 1.10 models the classes that describe the problem domain. The diagram depicts the generalization relationship of CH-Node and SEN-Node classeswith class 'Sensor_Node_Components'. There is a composition relationship between Sensor_Node_Components' and all the components deployed on the sensor board.

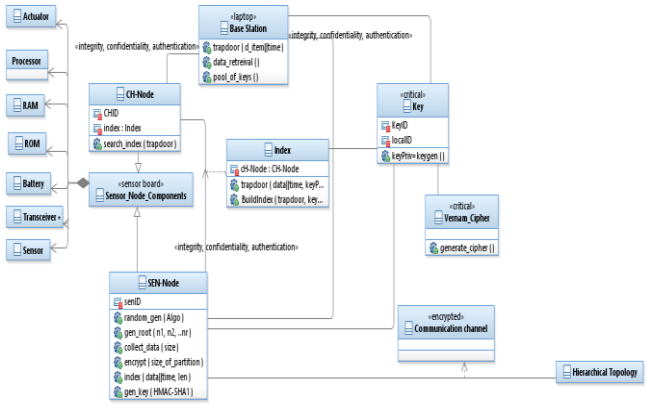

Figure 1-10 Class Diagram

The constraints that have been used on the classes have been deployed using UMLSec notations [20]. UMLSec is profile to develop security models in UML. It represents confidentiality, integrity and other security parameters using stereotypes, tags and constraints. Similarly, in the class diagram in fig 1.10, the stereotype <<critical>>has been used to depict significant security paradigms. Hence, classes 'Vernam_Cipher' and 'Key' are stereotyped as <<critical>>. The communication channel is an association class and is adorned with the tag <<encrypted>> as this implies that data will be encrypted and sent on the channel. The class diagram has used these annotations, depicting the semantics of the problem domain and even covering the hardware architecture.

IV. CONCLUSION

The unified modelling language has been used to create a structure for secure index on distributed data (SIDD) within the framework of unified development process.UMLhas assimilated the specifications of the application to make the design more modular and reusable. It is seen that UML can be efficiently used to design a heterogeneous network which deploys different kinds of nodes. The use case diagrams have been efficiently handled the basic constructs of a wireless sensor network application as demonstrated in the case of SIDD.

The formalization process has also been able to remove anomalies and redundancy in the existing algorithm bringing in a reliable application that can be deployed on actual nodes; even though the discussion of entire use cases is out of scope of this paper. The UML has been extended to apply to real world scenarios. The class diagram has modelled the software and hardware requirements of the application.The analysis of UMLSec [19] extensions has allowed to incorporate security paradigms and more profile additions would allow to construct more diagrams in the WSN scenarios.

This framework can be used to analyze the security paradigms of the application by introducing an adversary. With the structuring of this framework, introducing an adversary would require simple modifications to the existing design and evaluate the design for network throughput and energy consumption for future work.

REFERENCES

1. Bhasin, Vandana, P. C. Saxena, and C. P. Katti. (2018, July)"Creating a secure index for distributed data on the sensor network." *International Journal of Sensor Networks* 27.3: 180-199.
2. Parakh, Abhishek, and SubhashKak. "Online data storage using implicit security." *Information Sciences* 179.19 (2009): 3323-3331.

3. Losilla, F., Vicente-Chicote, C., Álvarez, B., Iborra, A., & Sánchez, P. (2007, September). Wireless sensor network application development: An architecture-centric mde approach. In *European Conference on Software Architecture* (pp. 179-194). Springer, Berlin, HeidelbergW.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
4. Mozumdar, M. M. R., Gregoretti, F., Lavagno, L., Vanzago, L., &Olivieri, S. (2008, June). A framework for modeling, simulation and automatic code generation of sensor network application. In *2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks* (pp. 515-522).
5. Selic, B. (2002, March). Complete High-Performance Code Generation from UML Models. In *Proceedings of Embedded System Conference*. San Francisco, CA, USA.
6. Rumbaugh, J. J. (1998). The Unified Modelling Language Reference Manual. Redwood City, CA, USA: Addison Wesley Longman Publishing Co
7. Uke, N. J., &Thool, R. C. (2015). Objects tracking in video: A object–oriented approach using Unified Modeling Language. *International Journal of Computational Vision and Robotics*, *5*(2), 202-216.
8. Infantino, I., Cossentino, M., &Chella, A. (2002, June). An Agent Based Multilevel architecture for Robotics Vision Systems. In *IC-AI* (pp. 386-390).
9. Gavrilescu, M., Magureanu, G., Pescaru, D., &Jian, I. (2012, November). Towards UML software models for Cyber Physical System applications. In *2012 20th Telecommunications Forum (TELFOR)* (pp. 1701-1704). IEEE.
10. Jacoub, J. K., Liscano, R., Bradbury, J. S., & Fisher, J. (2013). UML Modelling of Design Patterns for Wireless Sensor Networks. In *SENSORNETS* (pp. 89-93).
11. Fuchs, G., & German, R. (2010, May). UML2 activity diagram based programming of wireless sensor networks. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications* (pp. 8-13). ACM.
12. Hong, Sunghyuck, Sunho Lim, and Jaeki Song. "Unified Modeling Language based Analysis of Security Attacks in Wireless Sensor Networks: A Survey." *KSII Transactions on Internet & Information Systems* 5.4 (2011).
13. Pawar, P. M., Nielsen, R. H., Prasad, N. R., Ohmori, S., & Prasad, R. (2012). Behavioral Modeling of WSN MAC Layer Security Attacks: A Sequential UML Approach. *Journal of Cyber Security and Mobility*, *1*(1), 65-82.
14. Rodrigues, T., Dantas, P., Pires, P. F., Pirmez, L., Batista, T., Miceli, C., &Zomaya, A. (2011, October). Model-driven development of wireless sensor network applications. In *2011 IFIP 9th International Conference on Embedded and Ubiquitous Computing* (pp. 11-18). IEEE.
15. Uke, S., &Thool, R. (2016). UML based modeling for data aggregation in secured wireless sensor network. *Procedia Computer Science*, *78*, 706-713.
16. Specification, OMG Available. "Omg unified modeling language (omguml), superstructure, v2. 1.2." *Object Management Group* 70 (2007).
17. Jacobson, Ivar. *Object-oriented software engineering: a use case driven approach*. Pearson Education India, 1993.
18. Phillips, C., Kemp, E., &Kek, S. M. (2001). Extending UML use case modelling to support graphical user interface design. In *Proceedings 2001 Australian Software Engineering Conference* (pp. 48-57). IEEE.
19. Berardi, Daniela, Diego Calvanese, and Giuseppe De Giacomo. "Reasoning on UML class diagrams." *Artificial intelligence* 168.1-2 (2005): 70-118.
20. Schmidt, Holger, and Jan Jürjens. *UMLsec4UML2-adopting UMLsec to support UML2*. TU, Department of Computer Science, 2011.

AUTHORS PROFILE

Dr. VandanaBhasinis an Associate Professor at LalBahadurShastri Institute of Management, Dwarka, Delhi, since 2005. She is a PhD from Jawaharlal Nehru University, Delhi in the area of Security in Wireless Sensor Networks. Her main research interests include distributed systems, data security and sensor networks. Her teaching interests include Operating system, Linux, Object oriented analysis and design, data communication networks and mobile computing.