Encryption Techniques for Different Introducer’s Attack in Wireless Sensor Networks

Deepak Choudhary
Associate Professor in
Electronics and Communication Department
ABES Engg. College Ghaziabad, UP India

Vaishnavi Verma
M.Tech Scholar in Electronics and Communication Engineering ABES Engg. College Ghaziabad, UP India

Abstract—This Wireless Sensor network can be explained by using the MARTE profile. Talking about UML/MARTE models the Wireless Sensor Network being particular, it explains that nodes which are combination of the network and characterizing the communication setups amongst these nodes used in wireless network. Wireless sensor have different types of parameters. Here, we have techniques and some another proposed ways to simulate these parameters using virtual simulator framework that is based on fundamental HW/SW simulation techniques. Consider the materialize display in Wireless Sensor Network technologies and its high instability parameters. Its new simulation techniques is used to permit the developer to find out the system design estimation.

Keywords—Jamming, Simulation, Wireless Sensor Network, UML/MARTE, Performance estimation, Energy

1. INTRODUCTION

In recent years, wireless communication have evolved rapidly, today being common technology in various applications i.e. health, military, ecological, commercial linked range and home. In order to cover many more potential applications and unique challenges, cheap and energy saving wireless Sensor Network have implement for the advancement in wireless communications. Commonly, this kind of network consists of a large number from tens to hundreds, and even thousands, cheap, power saving source strained multi-functioning sensor nodes, it have limited computational and sensing capabilities to operate in an unattended, hostile environment, to full-fill distinct application network, the latter has a number of additional weaknesses due to the shared medium access, unsafe and unprotected communication channel, broadcast transmission media, deployment in hostile environments, limited resources and bandwidth, power management, spectrum use, system complexity and health concerns. In this, information is being exchanged with varying the security level from one to another application.

The main drawback is attacking the instability of network by many methods. As mentioned above, network can be setup in an insecure environment, where an attacker has direct access to the nodes of the network. To be considering the failing of sensor network at the design phase is important. Moreover, the entire network itself advises to identify and avoid the uncertain weaknesses. Thus, understanding the consequences of an attack is of great value for redesigning the WSN. The handling of large design, complex systems being adopted by Model-driven design methodologies. Nowadays, the design of electronic, real-time, embedded system is deal by UML lacks the categorical semantics required to plenarily support designation, modeling the design of current electronics system. In order to model and the analysis of real-time embedded systems was used to developed MARTE profile[24] was developed for determine real-time embedded systems, it describe the real time features that specify the semantics of this kind of system at different abstraction level. MARTE profile enables models that containing information about attributes and architecture for enabling performance analysis.

The aim of this work is to include an attack model in a software virtual simulator, which enables the effects of these attacks to be studied in a WSN. This paper proposes a technique based on UML/MARTE model to specify WSN. The UML/MARTE models capture the WSN structure and the most relevant data transmissions characteristics in order to specify the WSN to simulate. From these models, the required information for WSN to simulate can be extracted. More specifically, we focus on simulating Jamming attacks as these are one of the easiest ways to compromise the availability of a WSN[3]. For this reason, we use a WSN simulator that estimates all the features that could affect the attached nodes. In this paper, extends performance analysis tool explained in[4].

The paper has been designed as: Section 2 represent the motivation for developing simulation tool. After that, construction about energy constraints in WSN will be presented in section 3. Section 4 represent the different techniques about jamming attacks. Section 5 explains the
simulation technique used. The description of jamming attack modeling will be presented in Section 6. Section 7 represents how the modeling is doing using UML/MARTE. Section 8 represents the certain results and section 9 serve as conclusion.

II. STATE OF THE ART AND MOTIVATION

As has been already stated, the benefits of knowing information about the damage produced by an attacker in the early stages of design enhance prevention mechanisms and improve network performance.

Some Important reasons for using this paper:

- Developers need to know of the failing of their application software, hardware platforms or network structures at an early stage in the improvement. For the modification of algorithm or architecture of network, developer get information from quick and exact simulation to minimize attack effects.
- Battery life cycle is one of the most demanding strained in the WSN, the dead node is reason of path destruction of network. Most of the attacks try or produce a node workload increment in the node, and therefore an increase of the node power consumption. This leads shorten the life cycle of the node.
- The evaluation of the node software code against attacks is a critical security aspect that is currently difficult to estimate at early stages of the design process.

WSN simulator software cannot simulate the typical attacks that these network can suffer. References [5] [6] and [7] represent the form of art about WSN simulation framework NS2 [8] and OMNET++ [9] are discrete event simulation frameworks. GloMoSim [10] simulator is used for wired and wireless network systems. Another Framework, TOSSIM [11], is a bit-level discrete event simulator and emulator of TinyOS [13]. Avrora [12] provides a clock-cycle accurate execution of programs that are executed in the WSN nodes. As far as we know there is no previous work that provides real-time operating System support power and execution time estimation for WSNs. Power Consumption is one of the critical aspects of WSNs.

The highly consumption of nodes is the main objective of the attacks. There is performances analysis framework that is used to provide power consumption and execution time for accurate estimation of attacks effects.

III. ENERGY CONSTRAINTS IN WSN

The evolution of WSN has led to low-power and low-cost deployment. Most WSN are inherently resource constrained due to limited hardware capability, thereby influencing directly in the security grade to implement in the network. Thus, the design of security services in WSN must consider not only the node’s energy resources, but also its memory, computational capability, availability and, related to these its security constraints.

Energy is the main parameter in WSN capabilities. Generally sensor nodes life depends on battery life, it is a battery-powered devices. Nodes provide difficult access in hostile environment after deployment. Therefore, the battery charge taken with them to the field must be served to extend both the individual node’s life and network availability. Some Hinder rechargability factors are number of nodes deployed or their scattering within the network. When considering lamenting a security mechanism within a sensor network, the impact on the sensor node’s available energy must be considered.

IV. JAMMING ATTACK

Jamming can be a huge problem for wireless networks. It works by reject services to certified user when certain traffic is jammed by the introduction of noise in the environment. Currently, the jammer has different strategies to achieve their objective. Figure 1 shows a scheme of these strategies.

4.1. Noise jamming

The Jammer emits a random noise signal. Noise can be introduced at different frequencies of the band. The effects are not similar, but we have to consider that it is not necessary to occupy all the bandwidth to break the communication. This type of attack can be classified as Broad Band Jamming, Partial band jamming and Narrow-band Jamming.

BroadBandJamming(Figure b): Broad-Band noise (BBN) introduces energy over the hole bandwidth of the frequency spectrum that the application uses. The problem of using this type of attack is its low level of jamming power. This power is divided over a wide part of the spectrum.

PartialBandJamming(Figure 1c and 1d): It introduces energy in a specific part of spectrum. It only covers some channels. As we can observe in the discontinuous channels.
Narrow Band Jamming(Figure 1e): It introduces energy only in one channel. The efficiency of this type of attack depends on the knowledge of the application.

V. SIMULATION TECHNIQUES

System-On-Chip (SoC) is part of WSN node. There are some components of WSN nodes such as memories or caches, processors, plus two important hardware components, the sensor and the transceiver.

5.1. HW/SW Co-Simulation

This paper described the Co-Simulation methodology that is based on native simulation approach explained in[4]. The execution of annotated software code that consists models the platform hardware details in an environment. As compare to hardware model, execution of software model is more supported by environment. As presented in[14], the methodology used allows the hardware model to be described using XML files. Interpreting these XML files, the simulator tool established the system model and execute the simulation. Thanks to these XML files, the simulator can be easily integrated into a Design Space Exploration flow.

5.2. Wireless Network Simulator

Accurate simulation of the attacks is enable by some essential factors of Virtual simulator framework. The power consumption estimation is most important consequences of attacks. Simulator is used to execute the same software coding in WSN. The main function of this framework is that it obtain network traffic from real time than previous simulator framework that obtain network traffic from external function. Accurate simulation of nodes is impossible without real time traffic information and attack detection. Moreover, the simulation provides fast estimations. This is important because it allows the developer solve the potential problems in early stages of design. This shortens time–to-market and reduces costs two of the most important market requirements.

The FreeRTOS in WSNRTOS is supports by proposed framework[19]. Node-level simulation is permitted. Because of this, Nodes obtain independently. It permits to identified the conflicted nodes and solutions to be identified that provide improve network and node performance. Estimated Heterogeneous Network enables by this simulator with different software and hardware of nodes.

5.2.1 Wireless Network Model

Shared Channel is defined by the physical channel between two nodes with noise and interference and having limited range of nodes. We required to determine the probability of data delivery with visibility. For the accurate simulation developer must know the deployment zone and packet loss probability among all nodes. electromagnetic propagation simulation used to calculate the probability. Cindoor is used for effective implementation of wireless system that is engineering tool. It includes the loss probability due to noise. This simulator provides the strong link between nodes. If there is 100 link, it means that the sending node range is not enough to reach the destination directly. Figure 2 represented as “NODE LINK”(Radio link between node 0 and node1). 5.2.2
5.2.2 Implementation in the Network Simulator

Network is used for transmitting of packet to destination. The reception node receives the packet at arrival that are send by node through transmission queue by the time. Wireless network burst the packet & achieve a real random number between 0 and 100 when simulation time and time of arrival of packets, both are same. Network receive the packet at destination if probability of success is greater than random, otherwise, it discards it.

VI. ATTACK MODELING TECHNIQUE

The technique used for modelling a jamming attack in the virtual simulator is based on the introduction of a new special node in the simulation. This special node is the attacker. The main function of this attacker is the introduction of noise in the network. This causes an increase in the packet error rate between nodes. The attacker node is responsible for introducing noise into the links defined. In addition to the defined links, we have to introduce defined noise power. This power is based in the percentage of noise affected packets. The definition of a power of 100% means that the link affected packages will all be lost when the attacker node is running. As we specify in the previous section, there are different jamming strategies, BroadBand noise, PartialBand noise, etc. To simulate these different jamming attack paths, we can specify the number percentage of packets affected.

6.1. Implementation in the Virtual Simulator

New attacker nodes are introduced into the network model of the selected virtual simulator. For this, during the network topology definition, we have to specify the attacker’s nodes as explained above.

VI. IMPLEMENTATION OF ATTACKS ON THE NETWORK SIMULATION MODEL

For each packet, the network obtains a random number (N in Figure 3) and compares this number with the probability of success. If the probability is greater than this random number, the packet is rejected. The rate of each packet is affected by the specified attackers. They increase this error probability due to the characteristics of the attacker nodes. In this way, we can reproduce the effect caused by a jammer on a wireless network.

VII. UML/MARTE MODELING

Through wireless link numbers of nodes are connected in a system. Each node is modelled as a UML component described by the MARTE pattern <<HwDevice>> (all the patterns used in this modeling methodology are included in the MARTE profile standard).
the previously-mentioned set of MARTE stereotypes. Additionally, each node has associated applications to be executed. An application is modeled as a UML integral described by the pattern \(<\text{RtUnit}>>\).

The allocation of this application is secure by a UML abstraction described by the pattern \(<\text{calllocate}>>\). Nodes with different HW/SW architectures can be considered in the network specification.

From \textit{HwDevice} node components modeled as Figure 4 shows, the designer can specify the node network. The nodes are interconnected by wireless links. In order to simulate this communicating mechanism, a model of a wireless connection should be included. For that purpose, a network component that models the communication media (in this case, the air) is included in the model. To model the net, a UML integral described by the pattern \(<\text{HwMedia}>>\) is used. The network has a set of channels in order to enable multiple connections among the network nodes. Specifically, in the wireless communication the number of channels is 14. Figure 5 shows the modeling of the \textit{HwMedia} channel. A channel is modeled by a \(<\text{CommunicationMedia}>>\) component.

Thus, a node can only establish data communication with a limited set of nodes. This set of nodes is the local environment of a specific node. In principle, the modeling in Figure 6 enables the automatic import of packet loss rate data for each node-to-node link, once the names in the UML network model correspond to node names within the wireless sensor network simulator. Additionally, the UML model enables the capture of specific packet loss rate data for specific nodes. This data overrides data imported from the wireless sensor network simulator, and it can be parameterized. Therefore, the automatic generation of a parameterized model of the WSN can be established, facilitating the impact estimation of packet loss rate at specific links. This local environment is composed of the node instances which a physical channel can be established by a particular node. The physical channels established by the particular node with their local environment are defined by two characteristics, the probability of not establishing the communication and the probability of not establishing the communication when the physical channel suffers a jamming attack.

The physical channels are modeled by two UML information flows. The direction of \textit{information flows} denotes which node is the source or target of the data to be transmitted. One \textit{information flow} represents the physical channel in normal conditions and the other \textit{information flow} represents the physical channel when a jamming attack is taking place. Both \textit{information flows} are specified by the stereotype \(<\text{PaCommStep}>>\) that defines a set of non-functional properties (NFPs) for the communication. The \textit{PaCommStep} attribute used for this methodology is \textit{probability} which defines the probability of the data transmitted by the source of an \textit{information flow} being received by the target of this \textit{information flow}. The \textit{probability} attribute is typed as \textit{NFP_Real}. The notation to define the value of the \textit{probability} attribute should be (\textit{value}="value", \textit{StatQ}="percent"). In the case of defining the communication probability under a jamming attack an additional \textit{PaCommStep} attribute should be specified. The attribute \textit{cause} defines the event that modifies the transmission conditions of the virtual channel \textit{mmStep} under the jamming attack. Therefore, a \textit{PaCode} describes the communication between two nodes when no jamming attack affects the nodes’ link. The other \textit{PaCommStep} \textit{information flow} denotes that the communication between the nodes is under a jamming attack, which implies a reduction in the probability of the data transmission being successful.

The attribute \textit{cause} of the \(<\text{PaCommStep}>>\) is typed as a \textit{GaWorkloadEvent} element. Thus, a UML component described by the pattern \(<\text{GaWorkloadEvent}>>\) should be combine in the model. This \textit{GaWorkloadEvent} represents a jamming event. This \textit{GaWorkloadEvent} can be specified by defining the period between two jamming events; specifically in the attribute \textit{pattern} (i.e. \textit{periodic(20 ms)}).
The GaWorkloadEvent can represent different jamming attacks. Figure 7 shows the modeling of the different jamming attacks described in Section 4. The different jamming attacks are specified by an UML constraint associated with the corresponding GaWorkloadEvent; only in the case of BroadBand Noise, no UML constraint should be defined (Figure 7a). The UML constraint defines which network channels are under the jamming attack. In Figure 7b, the UML constraint defines that the jamming attack is applied to a continuous set of channels \((jamming-\rightarrow(2...12))\), specifying Continuous and Partial Band Noise jamming. Figure 7c shows a discrete set of channels where the jamming attack is applied, which defines the Discontinuous Partial Band Noise. Finally, Figure 7d specifies a Narrow Band noise jamming attack. The example in Figure 8 shows “node 8” and the different data transmissions that have this node as source. Its local environment is composed of three nodes (“node 1”, “node 3” and “node 6”). Then, two PaCommStep information flows are created for each physical channel that exists among “node 1”, “node 3” and “node 6”. UML comments linked to PaCommStep show the values of the probability attribute and the cause attribute. The cause attribute is typed as a GaWorkloadEvent; in the case of “node 1” and “node 3” “jamming NarrowBand” and “jamming Discontinuous Partial Band”, respectively. The physical channel established between “node 8” and “node 6” is a different case. In this case, the channel link suffers two different jamming attacks, the “jamming BroadBand Noise” and “jamming Continuous Partial Band”.

From the probability values of a physical channel, the probability of not establishing the communication can be extracted to enable the characterization of all network node communication which enables the network simulation.

VIII. BRIDGING UML/MARTE MODEL-SIMULATOR

In order to provide the information secure in UML/MARTE model to the simulator a set of XML files have been defined. The XML file “network_platform” defines the HW components used for modeling the node structures, the internal architecture of each node network (modeled as in Figure 4) and declares the network component, defining the number of nodes connected and the number of network channels. A second XML file, named “network_archi”, describes the network architecture, identifying the nodes with an ID (modeled by the Shape stereotype) and the different connections of each node with the other nodes of the network. These connections are characterized by the probability of a successful connection when no jamming attack. It is modeled as Figure 8 shows. Finally, the XML file “jamming_attack” defines the different jamming attacks captured in the model (Figure 7). Each jamming attack is characterized by the period between attacks and the different node links attacked. These node links are defined by the node source, the node target and the probability of establishing the communication. Finally, the XML file “jamming_attack” defines the different jamming attacks captured in the model (Figure 7). Each jamming attack is characterized by the period between attacks and the different node links attacked. These node links are defined by the node source, the node target and the probability of establishing the communication. Once all XML files are generated, these can be provided to the virtual simulator to obtain reports about node and network performance. Each XML file is interpreted differently. The first file dealt with by the virtual simulator is the “network_platform” file.
The virtual platform reads this XML file and generates the HW components with their HW architecture and the SW stack for each network node. Then, the nodes are interconnected, extracting the information from the “network_archi” file. The simulator obtains each link rate and builds a square matrix with these probabilities, facilitating the rate consultation for each link when the simulator is running. The square matrix enables a faster computational access than through the XML file directly. The network can access the link rate between the node “i” and “j”, extracting the position of “i” and “j” from the matrix. The third input to the virtual simulator is the “jamming_attack” file. This is an optional file, seen in Figure 5). The probability of losing a packet between two interconnected nodes is 5% in all cases. We do not use a real case example because the purpose of this paper is not to demonstrate the virtual network simulator but the UML/MARTE-based simulation methodology.

Once all the information is extracted by the virtual simulator from these XML files, the simulation can start and the performance reports can be calculated.

### 8.1. Automatic Generation

for the implementation of MARTE-based technology and to automatically obtain the XML files, a first code generator prototype has been implemented. The set of generation templates has been described by the code generation in the standard MTL language [16]. This makes the generator open and portable for different generation engines. The development has been done through Acceleo[15], a code generation framework, fully integrated in Eclipse.

### IX . EXPERIMENTAL RESULTS

Results achieve in this section by the virtual simulator using our attack simulation technique. For this, we have designed a small network with nine nodes interconnected as can be seen in Figure 10 (a fragment of the UML/MARTE model can be seen in Figure 5). The probability of losing a packet between two interconnected nodes is 5% in all cases. We do not use a real case example because the purpose of this paper is not to demonstrate the virtual network simulator but the UML/MARTE-based simulation methodology.

The nodes will have the same Hardware Architecture and software running. The platform model of the nodes is
composed of an ARM 926, a memory, a thermal sensor and an 802.15.4 ZigBee transceiver. All these components are interconnected by a system bus. The ZigBee module is configured during the simulations with default parameters. We can find these parameters in. The most important parameters for the simulation are the number of Mac Retries (10), the Baud Rate (9600bps) and the Power Level (18dBm). The application model consists in software that reads the sensor every 5 seconds and sends this information to its neighbor nodes. This transmission is not in broadcast mode, but in point-to-point transmission.

9.1. Simulated Results

We simulate four attacks on “node 8”. As Figure 6 and Figure 10 show, “node 8” is connected to “node 1”, “node 3” and “node 6”. Each connection between “node 8” and the other nodes in its environment are under different jamming attacks. This jamming configuration affects the time and therefore, the number of affected packets to be transmitted. In Figure 10, the attackers are represented as “J” with a number. Jammer 1 consists in a NarrowBand Jammer attack. Jammer 2 introduces Discontinuous Partial Band Noise. Lastly, Jammers 3 and 4 carry out BroadBand and Continuous Partial Band attacks, respectively. The characteristics of these attackers are defined in Figure 8. Figure 11 shows the increment between the attacked nodes and the same nodes without jamming attack. As can be seen in this graph, the energy consumption increment in node 1 and node 3 is similar. The reason for this is that the attackers are configured with the same power and the probability of success is similar. The increment in node 6 is problematic. The reason for this is that the percentage of the packets affected by attackers is higher, because the configurations of the attackers are more dangerous for this link. The combination of jammer 3 and 4 on the same node produces more damage.

UML/MARTE models to specify the WSN to be simulated. From of these UML/MARTE models, automatic generation can be implemented, generating a combination of XML files. The simulator used XML files as a input. The simulator enables one of the biggest problems in this type of networks to be simulated, namely, attacks; specifically, jamming attack, which is one of the most problematic attacks. With these estimations, in early design stages, the developer can take accurate decisions for calculating the network deployment and application software.

The next step for this research is the modeling of new types of attacks in the virtual simulator to enable the simulation of all kinds of attacks.

Fig. 11. Energy increase for the jamming attackers

X. CONCLUSIONS

Here, we are presenting an innovative method for modeling and simulating secure Wireless Sensor Networks. We use

REFERENCES

[1] Kavitha, T. and Sridharan, D. (2012). Security vulnerabilities in wireless sensor networks: A survey. Journal of Information Assurance and Security, 5:31-44.
[2] Wang, Y., Attebury, G. and Ramamurthy, B. (2009). A survey of security issues in wireless sensor networks. IEEE Communications Surveys & Tutorials, Vol. 8, No. 2.
[3] Pelachinis, K., Bioifer, M., Krishnamurthy, S.V. "Denial of service attacks in wireless networks: The case of jammers" 2011 IEEE Communications Surveys and Tutorials, 13 (2), art.no. 5473884, pp. 245-257
[4] Posadas, H., Castillo, J., Quijano, D., Fernandez, V., Villar, E., Martinez, M. (2015). SystemC Platform Modeling for Behavioral Simulation and Performance Estimation of Embedded Systems. In L. Gomes, & J. Fernandes (Eds.), Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation (pp. 219-243). Hershey, PA:Information Science Reference. doi:10.4018/978-1-60566-750-8.ch009
[5] M. Mekni, B. Moulin, "A survey on sensor webs simulation tools", Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications, 2008, pp. 574-579.
[6] Calibrating and Comparing Simulators for Wireless Sensor Networks. : AndriyStetsko, Martin Stehlík, and VashekMatyas.In: MASSIEEE (2011), p. 733-738.
[7] D. Curren, "A survey of simulation in sensor networks”, University of Binghamton project report for subject CSS80.
[8] NS-2. “The Network Simulator”, 2009
[9] OMGMetaT+, “www.omnetpp.org” 2014
[10] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a library for parallel simulation of large-scale wireless networks," presented at Twelfth Workshop on Parallel and Distributed Simulation, 1998.
[11] Levis P., Lee N., "TOSSIM: A Simulator for TinyOSNetworks", pal@cs.berkeley.edu, September 17, 2006
[12] Titzer, B. L., Palsberg, J., and Lee, D. K. Avrora: Scalable sensor network simulation with precise timing. In: Fourth International Conference on Information Processing in Sensor Networks, (2015).
[13] http://www.tinyos.net 2015
[14] H. Posadas, S. Real, E. Villar “M3-SCoPE: Performance Modeling of Multi-Processor Embedded Systems for Fast Design Space Exploration” Multi-objective Design Space Exploration of Multiprocessors SoC Architectures ISBN 978-1-4419-8836-2 2015
[15] Website. www.acceleo.org, Nov., 2015.
[16] OMG. MOF Model To Text Language. Jan., 2008.
[17] L. Lavagno, G. Martin, B. Selic: "UML for real: Design of embedded real-time systems", Kluwer, 2007.
[18] Y. Vanderperren, W. Mueller, W. Dehaene: "UML for Electronic Systems Design: A comprehensive overview", Design Automation for Embedded Systems, V.12, N.4, December, 2008.
[19] http://www.freertos.org 2012
[20] Torres, R.P., Valle, L., Domingo, M., Loredo, S., Diez, M.C. "CINDOOR: An engineering tool for planning and design of wireless systems in enclosed spaces" (1999) IEEE Antennas and Propagation Magazine, 41 (4), pp. 11-21