Network Intrusion Detection System for Jamming Attack in LoRaWAN join procedure

Syed Muhammad Danish*, Arfa Nasir*, Hassaan Khaliq Qureshi*, Ayesha Binte Ashfaq*,
Shahid Mumtaz† and Jonathan Rodriguez†
*National University of Science & Technology (NUST), Islamabad 44000, Pakistan
*{sdanish.msee16seecs, anasir.msee16seecs, hassaan.khaliq}@seecs.nust.edu.pk, *aashfaq@pnec.nust.edu.pk
†Instituto de Telecomunicaes, Aveiro, Portugal
†{smumtaz, jonathan}@av.it.pt

***Abstract*—LoRaWAN is a Low Power Wide Area Network (LPWAN) protocol designed to allow low power battery operated nodes to communicate with each other. Though LoRaWAN provides end-to-end security, however vulnerabilities exist in the security mechanism of LoRaWAN join procedure. A jammer can be used to launch a denial of service (DOS) attack by permanently disconnecting the LoRa end nodes from the LoRaWAN network. In this paper, we propose a novel LoRaWAN based Intrusion Detection System (LIDS) for jamming attacks. A real experimental testbed is developed and deployed and LIDS is trained on real join request data. We propose two LIDS algorithms based on Kullback Leibler Divergence (KLD) and Hamming distance (HD). The algorithms are extensively tested on real-world dataset. Receiver Operating Characteristic (ROC) based performance evaluations show that KLD and HD can achieve detection rates as high as 98% and 88% respectively with 5% false positive rate.**

I. Introduction

According to Gartner [1], there will be 20.4 billion IoT devices by 2020. Consequently, there is a rise on the spending on IoT services both in the consumer market as well as in industry. Presently, IoT is gaining major deployment for applications like smart homes, smart cities, agriculture, farming, emergency and security, transportation & environment monitoring etc. IoT devices have limited memory, low processing capability and less energy resources so connecting them to internet is challenging. Also, sensitive information regarding companies and people is handled by IoT applications which should be secured and only accessible to authorized people.

Recently, low power wide area networks (LPWAN) have been deployed to cater for the specific needs of power constrained IoT devices. LoRaWAN is an example of LPWAN and it provides features such as low power, mobile and secure bidirectional communication between IOT devices. LoRaWAN is especially compatible with low power, battery operated devices which are becoming more common. For security, LoRaWAN is designed to provide integrity, authentication and confidentiality for secure transmission and provide end-to-end encryption at application layer and integrity at network layer. However, LoRaWAN is currently at its beginning stage and much work has yet to be done in making LoraWAN better and more secure.

LoRaWAN is a new technology and hence some security features of the protocol are yet to be properly defined. LoRaWAN has been shown to be susceptible to replay attacks [2][4], jamming attacks [2][7], wormhole attacks [2] and bit flipping attacks [3]. Authors in [7] explain the vulnerability of LoRaWAN protocol to jamming attacks, also known as intentional interference attacks. These attacks lead to denial of service by limiting the ability of a LoRa end nodes to connect to the LoRaWAN network. Authors in [8][9][10] explain the detection and mitigation techniques for jamming attacks in Wireless Sensor Network (WSN) based on percentage of incurred collision, signal strength, packet delivery ratio etc. However, all these techniques are based on data communication. Since no data communication occurs before a LoRaWAN join procedure, so a mechanism is required to detect jamming attack in LoRaWAN join procedure.

In this paper, we propose an intrusion detection system for the detection of jamming attacks in a LoRaWAN network. We implement a real testbed based on LoRaWAN protocol. The testbed allows LoRa end nodes to send join requests to network server through gateway and in turn receive the response from the network server through gateway. A jamming attack is performed with the help of LoRa SX1272 mbed shield. To the best of our knowledge, this is the first intrusion detection solution for detection of jamming attacks in a LoRaWAN network.

A hardware testbed has been implemented in which arduino with LoRa mbed shield is used as LoRa end node, raspberry pi is used as a gateway and a network server running on a laptop. LoRa end nodes are placed at a distance of few meters and jammer is placed at a distance of approx. 1m. LoRaWAN, unlike previous IoT technologies, provides long range communication; 15Km for Line of Sight (LOS) and 5Km for urban area [11]. The LoRa end nodes are deployed far away from the gateway and network server. Jammer, when deployed in the vicinity of LoRa end nodes, can affect the normal transmission as well as cause a denial of service attack [2]. The LIDS algorithms are deployed on the gateway to monitor the real-time traffic patterns of the join request transmission from LoRa end nodes. The real time traffic distribution is compared with the baseline distribution for the detection of the jamming attack. We show that the LIDS algorithms are able to achieve high detection rates while minimizing the false positives.

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

The rest of the paper is organized as follows. Section II provides an overview of LoRaWAN join procedure and how few LoRa end nodes can be compromised by use of Jammer. We then propose LIDS algorithms and their working in Section III. Section IV gives a brief overview of our hardware experimental setup. Section V and VI summarize the performance and key conclusions of this paper.

II. LoRaWAN join procedure

In LoRaWAN network, each LoRa end node has to perform a join procedure in order to connect to LoRaWAN network. In Over-The-Air-Activation mode (OTAA), a join request message is sent by LoRa end node and if this join request is valid, join accept message is generated by the network server. This join request message has three components: (1) AppEUI (8 Bytes); (2) DevEUI (8 Bytes); and (3) DevNonce (2 Bytes). DevNonce is a 16 bit random number and it is generated by N read operations of the least significant bit of the register RegRssiWideband and the corresponding address of this register is 0x2c. The value from this register is obtained from the wideband (4 MHz) signal strength at the receiver input every 1ms. It is also assumed that the LSB constantly and randomly changes due to the noise and radio channel behavior (reflection, fading, shadowing and interference) [12].

Every time a LoRa end node wants to join a network, it generates a new 16-bit DevNonce for a join request. If in case a previously used DevNonce is used, network server can perform 2 actions:

- the network server will drop the join request
- the network server will disconnect the LoRa end node for which join request is received

Jamming attack can be launched either by transmitting at the power higher than received power maximum value or by transmitting at high and constant received power so the quantized Received Signal Strength Indicator (RSSI) value become constant in time[1]. The jamming attack will result in a DOS attack since the LoRa end nodes in the vicinity of the Jammer will not be able to join or access the LoRaWAN network. Moreover, in a LoRaWAN network, the random number generator does not have a health check [13], which makes it more vulnerable to attacks [7]. A brief overview of proposed LIDS algorithms is given in Section III.

III. LoRaWAN based Intrusion Detection System (LIDS)

We now explain in detail the KL-based and HD-based algorithms used for detection of jamming attacks in our LoRaWAN network setup. Later we provide an ROC-based analysis of the performance of these algorithms.

A. Kullback Leibler Divergence (KLD) based LIDS

Kullback Leibler divergence (KLD) [15] is a directed divergence between two distributions. It simply implies how much one distribution diverges from another. Phrased differently,

[1]Readers are referred to [7] for more details.

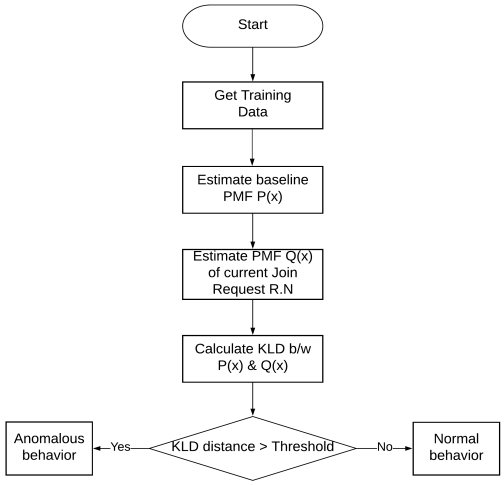

Fig. 1. Working of KLD based algorithm

KLD compare the statistical similarity of two distributions. If we have two distributions P and Q, a smaller value of KLD implies more similarity/less divergence and vice versa.

Hence, given two probability mass function P and Q, the KLD can be calculated as:

$$D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

The KLD based LIDS learns the distribution of the 16-bit random number generated by the LoRa node during join request. This learnt distribution is the baseline distribution P(x)[2] where $x \in [0, 2^{16}]$. Q(x) is the real-time distribution of the random number in the join request. KLD based LIDS computes the divergence between P(x) and Q(x) in realtime. If the divergence exceeds a threshold it is termed anomalous, otherwise it is normal. Divergence between P(x) and all the possible distributions of Q(x) is calculated and threshold for KLD based LIDS is selected, based on this divergence. An illustration of the working of the algorithm is given in Figure 1.

B. Hamming Distance based LIDS

Hamming distance [14] depicts the difference between two binary vectors. Given vector A & vector B of equal length, Hamming distance between vector A & B can be calculated by counting the position in which corresponding bits of two binary vectors are different.

The Hamming distance W_d between A and B can be calculated as:

$$W_d = \sum_{i=1}^{n} \delta(A_i \oplus B_i)$$

[2]Our P(x) is approximately the same as calculated in [7] with slight difference owing to receiver saturation.

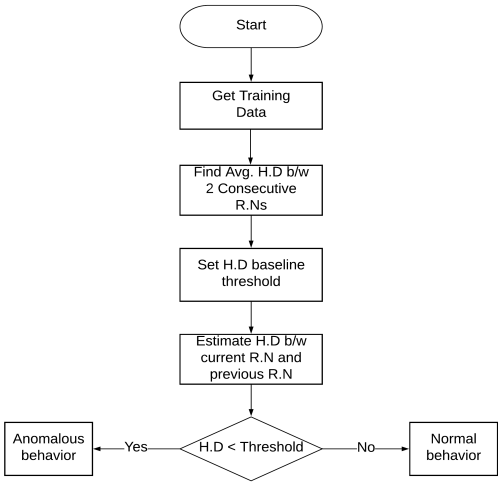

Fig. 2. Working of Hamming distance based algorithm

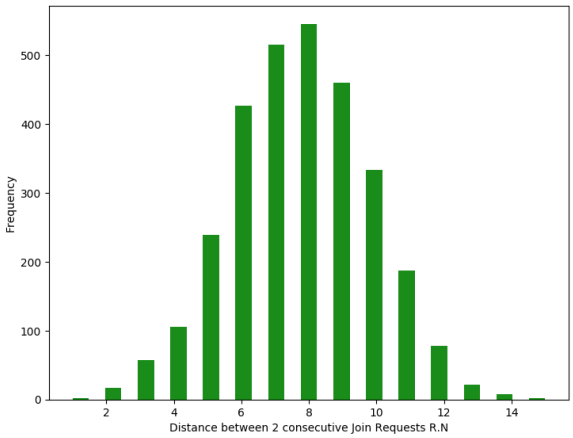

Fig. 3. Frequency Analysis of Hamming Distance between 2 consecutive join request random numbers

Where $\oplus$ represents the XOR operation between corresponding bits of A & B. Figure 2 illustrates the working of the HD based LIDS algorithm.

KLD based LIDS computes the baseline distribution, which is then compared with the run-time random number for attack detection. However, HD based LIDS is distinctively different. In this algorithm, we learn the baseline hamming distance between consecutive random numbers. This learnt distance is then used as a threshold to detect the presence of a jammer in the vicinity of a LoRa end node. Moreover, join request is signed by AppKey to ensure the integrity. Attacker needs the AppKey to correctly calculate the Message Integrity Check (MIC) for join request to know the baseline HD, which is very difficult to get. To launch the DOS attack, attacker needs the same DevNonce everytime. Even if the baseline HD is known, threshold will still be unknown to attacker and jammer will be detected by HD based LIDS because of small hamming distance between previous and current DevNonce.

The learnt baseline hamming distance is a characteristic of the random number generator. Figure 3 presents a histogram of the hamming distances between consecutive random numbers in our dataset. As can be seen, the histogram exhibits Gaussian behavior with a mean (μ) value of 7.75 and standard deviation (σ) of 2.34. These values are an attribute of the random number generator. The baseline threshold is computed as μ - σ. The μ + σ values are ignored. HD based LIDS offers best performance for threshold between μ - σ & μ - 2σ. We now explain our experimental setup in section IV.

IV. Experimental Setup

In our experimental testbed, arduino UNO with SX1272 mbed shield is used as a LoRa end node, raspberry pi 2 model B with Raspbian Jessie OS is used as a LoRa gateway and a network server has been written in python on a laptop with core i3, 6Gb RAM, fast ethernet LAN, running Linux Ubuntu OS as shown in Figure 4. Raspberry pi gateway is connected to network server with the help of ethernet cable. Server-side program is running on Laptop network server and Client-side program is running on Raspberry pi Gateway. Four LoRa end nodes have been used in this experimental setup and at each LoRa end node, SX1272 LoRa mbed Shield is used for LoRa RF Communication. Two other LoRa end nodes are placed on different locations as shown in Figure 6. Join procedure experiment has been performed to evaluate the performance of join request random number vulnerabilities.

Testbed topology is shown in Figure 5. LoRa end nodes are placed approximately 50m away from each other without direct line of sight with major obstacles in the way. Jammer is placed at a distance of approximately 1m from the main LoRa end nodes. Because of limited number of LoRa RF modules, communication between Arduino and Raspberry pi is done through serial communication. Irrespective of the location of LoRa end nodes, Over the Air Activation (OTAA) LoRaWAN join procedure remains the same.

Programming of Arduino in LoRa end node is done in such a way that it generates a join request after every 30 seconds. This is done mainly to collect more data. The testbed was setup in our lab and data has been collected for 2:30 hours each day over a span of 10 days.

V. Results

In this Section, we have investigated the performance of Hamming distance based LIDS and KLD based LIDS presented earlier for detecting jamming attack in LoRaWAN Network. In the first subsection, we explain the difference between the normal behavior and jamming attack behavior. Similarly, we present the performance of the two LIDS algorithms (in the form of ROC curves) in the next subsection.

A. Normal Behavior & Jamming Attack Behavior

We analyze the behavior of the random numbers during normal LoRaWAN operation as well as when a jamming attack

Fig. 4. Experimental setup in Lab 1. Network Server, Gateway, LoRa end node & Jammer

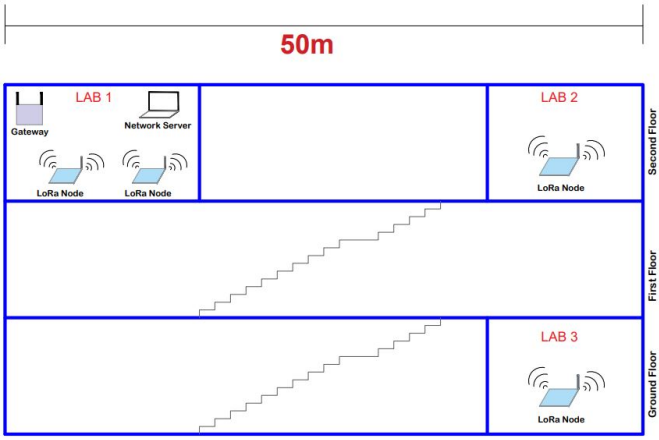

Fig. 5. Overall Testbed Setup

is launched. As already stated, these random numbers are sent from the LoRa end nodes to the network server as part of the join request packet. The analysis is performed on the hamming distance between consecutive random numbers. This hamming distance can be calculated as follows:

$$HammingDistance = W_d(R_1^{(n)}, R_2^{(n)})$$

where R_1 and R_2 are consecutive random numbers. These can be represented in terms of their corresponding binary bits as follows:

$$R_1^{(n)} = b_0^{(n)}$$

$$R_2^{(n)} = b_0^{(n)}$$

Where b_0 represents the LSB and n represent N read operations of RSSI value. These binary values are obtained from the decimal number as follows:

$$D = \sum_{i=0}^{n-1} 2^i b_i$$

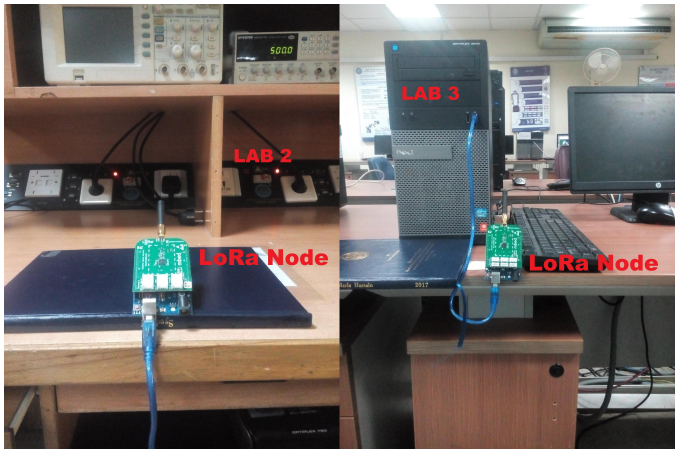

Fig. 6. Experimental setup in Lab 2 & Lab 3. LoRa end nodes

where D is number in Decimal format and b_i is corresponding binary bit for 2^i.

Let us first analyze the random numbers during normal LoRaWAN operation. This behavior is illustrated in Figure 7. The figure presents the hamming distance between consecutive random numbers received at the network server from a specific LoRa end node. Since the numbers are generated by a random number generator, hence the difference between these random numbers is also random to some extent.

We now analyze the behavior of the random numbers when a jamming attack is launched on the LoRa nodes in the LoRaWAN network. The jammer is just another LoRa end node based on Arduino with SX1272 mbed shield, transmitting at a constant power in the vicinity of LoRa end node and forcing LoRa end node to generate approximately constant RSSI value in RegRssiWideband register. The jammer is turned on for approximately an hour for join requests starting from 175 to 300. A dip in the hamming distance can be clearly seen in Figure 8. As jammer will try to keep RSSI value constant in RegRssiWideband register, approximately same random numbers will be generated and the hamming distance between them will decrease.

1) Discussion: Since jammer transmits with high constant power in the vicinity of LoRa end node, it forces RegRssiWideband register to write approximately constant RSSI value. Hence, during a jamming attack, the jammer forces to keep the RSSI values constant. Consequently the join requests will have approximately similar binary strings and random numbers $R_1^{(n)}$ & $R_2^{(n)}$. Since the random numbers are approximately the same, their hamming distance will decrease explaining the sudden fall in the hamming distance in Figure 8.

B. ROC Evaluation

Figure 9 presents the ROC curve for KLD based and HD based LIDS. Both algorithms provide comparable detection rates and false alarm rate for some thresholds. However after attaining 84% detection rate and 2% false alarm rate, both the algorithms part ways with KLD based LIDS providing higher detection rates than HD based LIDS. The ROC curves show

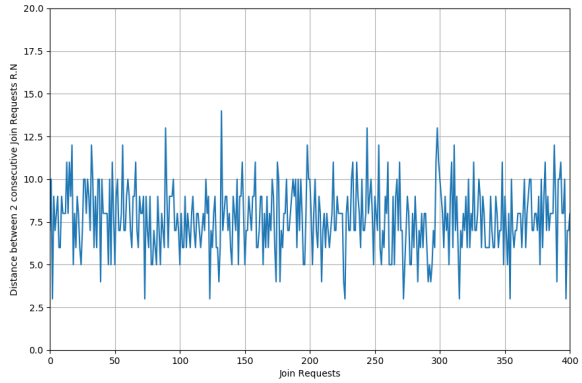

Fig. 7. Join request random number during normal operation

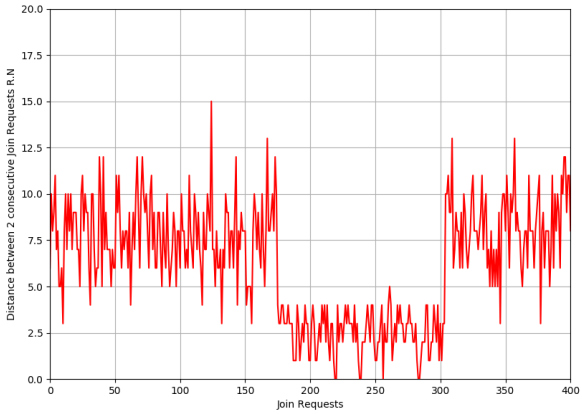

Fig. 8. Join request random number during jamming attack

the best operating point for KLD and HD are 98% and 88% detection rates respectively with 5% false alarm rate. Hence, KLD outperforms HD based LIDS. From figure 7, we can see that even in normal case sometimes hamming distance between consecutive DevNonce is small and this leads to generate false alarm in Hamming distance based LIDS. Also for some join requests, DevNonce doesn't strictly follow P(x), which leads to false alarm generation in KLD based LIDS.

VI. Conclusion

In this paper we have shown an experimental testbed for LoRaWAN join procedure and launched a jamming attack on this LoRaWAN testbed. We have investigated two LIDS algorithms named KLD based LIDS & Hamming distance based LIDS on the LoRaWAN testbed. From results we have seen that even a very simple Hamming distance based LIDS algorithm can provide satisfactory results. However, KLD based LIDS algorithm outperforms the previous algorithm and its detection rate is much higher keeping the false alarm rate constant. In future we are planning to implement multiple LoRa end nodes on testbed and perform experimentations to measure correlation between each LoRa end node join request random number.

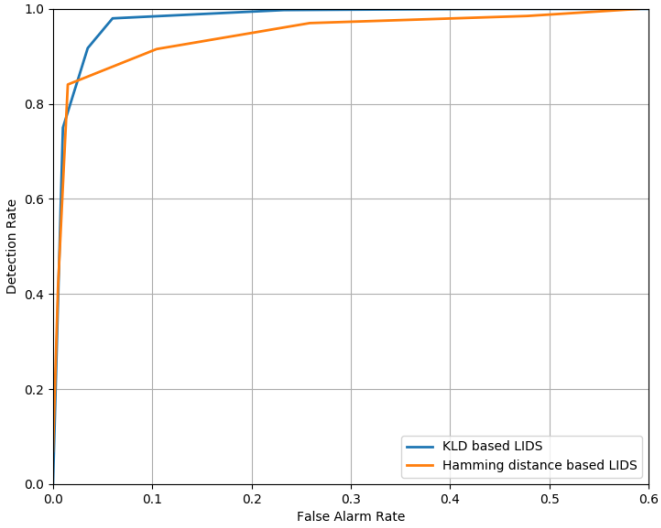

Fig. 9. ROC Curve for KLD based LIDS algorithm

Acknowledgment

The research leading to these results received funding and support from national funds through FCT/MEC (UID/EEA/50008/2013)

References

[1] "Gartner Says 20.4 Billion Connected "Things" Will Be in Use in 2020", 2017. [Oline]. Available: https://www.gartner.com/newsroom/id/3598917

[2] Emekcan Aras, Gowri Sankar Ramachandran, Piers Lawrence and Danny Hughes, *Exploring The Security Vulnerabilities of LoRa*: International Conference on Cybernetics (CYBCONF), 2017.

[3] JungWoon Lee, DongYeop Hwang, JiHong Park, and Ki-Hyung Kim, *Risk Analysis and Countermeasure for Bit-Flipping Attack in LoRaWAN*: International Conference on Information Networking (ICOIN), 2017.

[4] SeungJae Na, DongYeop Hwang, WoonSeob Shin, and Ki-Hyung Kim, *Scenario and Countermeasure for Replay Attack Using join request Messages in LoRaWAN* International Conference on Information Networking (ICOIN), 2017.

[5] Sarra Naoui, Mohamed Elhoucine Elhdhili, Leila Azouz Saidane, *Enhancing the security of the IoT LoraWAN architecture*: International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), 2016.

[6] H. Kopka and P. W. Daly, *Using Blockchain Technology to Build Trust in Sharing LoRaWAN IoT*: International Conference on Crowd Science and Engineering, 2017.

[7] Stefano Tomasin, Simone Zulian and Lorenzo Vangelista, *Security Analysis of LoRaWANTM Join Procedure for Internet of Things Networks*: IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2017.

[8] Mingyan Li, Iordanis Koutsopoulos and Radha Poovendran, *Optimal Jamming Attacks and Network Defense Policies in Wireless Sensor Networks*: IEEE International Conference on Computer Communications (INFOCOM), 2007.

[9] Wenyuan Xu, Wade Trappe, Yanyong Zhang and Timothy Wood, *The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks*: MobiHoc 05: Proc. 6th ACM Intl. Symp. Mobile Ad Hoc Net. and Comp., 2005, pp. 4657.

[10] A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C. Wong, *Decentralized Intrusion Detection in Wireless Sensor Networks*: 1st ACM international workshop on Quality of service & security in wireless and mobile networks (Q2SWinet 05). ACM Press, October 2005, pp. 1623

[11] https://www.digikey.com/en/maker/blogs/introduction-to-lora-technology-the-game-changer/99c9e3676f0a47339b5c0306af529242

[12] http://www.semtech.com/images/datasheet/an1200.24_ag.pdf.

[13] E. Barker, J. Kelsey, *Recommendation for the Entropy Sources Used for random Bit Generation*, NIST DRAFT Special Publication 800-90B, Second Draft, January 2016. [Online]. Available: http://csrc.nist.gov/publications/drafts/800-90/sp800- 90b_second_draft.pd.

[14] Yan-Heng Liu, Da-Xin Tian, Ai-Min Wang, *ANNIDS: intrusion detection system based on artificial neural network*: International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693), 2003.

[15] T. Cover, J. Thomas, and J. Wiley, *Elements of information theory*, ed. Wiley-Interscience, 2006, vol. 1.

[16] R. Miller, *Lora security: Building a secure lora solution*, MWR Labs, RFC, 2016. [Online]. Available: https://labs.mwrinfosecurity.com/assets/BlogFiles/mwri- LoRa-security-guide-1.2-2016-03-22.pdf

[17] F.Sabahi and A.Movaghar, *Intrusion Detection: A Survey* :The Third International Conference on Systems and Networks Communications, 2008.

[18] Afef Mdhaffar, Tarak Chaari, Kaouthar Larbi, Mohamed Jmaiel and Bernd Freisleben, *IoT-based Health Monitoring via LoRaWAN*: International Conference on Smart Technologies, 2017.

[19] Lu Tan, Neng Wang, *Future internet: The Internet of Things*: International Conference on Advanced Computer Theory and Engineering (ICACTE), 2010.